FORE
SYSTEMS®

PowerHub
Software Reference Manual

MANU0167-01 Rev A - November 7, 1997

Software Version PH_FT 4.0.0

FORE Systems, Inc.

1000 FORE Drive
Warrendale, PA 15086-7502
Phone: 412-742-4444
FAX: 412-742-7742

http://www.fore.com

Legal Notices

U.S. Government Restricted Rights. If you are licensing the Software on behalf of the U.S. Government ("Government"), the following provisions apply to you. If the Software is supplied to the Department of Defense ("DoD"), it is classified as "Commercial Computer Software" under paragraph 252.227-7014 of the DoD Supplement to the Federal Acquisition Regulations ("DFARS") (or any successor regulations) and the Government is acquiring only the license rights granted herein (the license rights customarily provided to non-Government users). If the Software is supplied to any unit or agency of the Government other than DoD, it is classified as "Restricted Computer Software" and the Government's rights in the Software are defined in paragraph 52.227-19 of the Federal Acquisition Regulations ("FAR") (or any successor regulations) or, in the cases of NASA, in paragraph 18.52.227-86 of the NASA Supplement to the FAR (or any successor regulations).

Printed in the USA.

No part of this work covered by copyright may be reproduced in any form. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

This publication is provided by FORE Systems, Inc. "as-is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose. FORE Systems, Inc. shall not be liable for any errors or omissions which may occur in this publication, nor for incidental or consequential damages of any kind resulting from the furnishing, performance, or use of this publication.

Information published here is current or planned as of the date of publication of this document. Because we are improving and adding features to our products continuously, the information in this document is subject to change without notice.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 (October 1988) and FAR 52.227-19 (June 1987).

The VxWorks software used in the Mini Loader is licensed from Wind River Systems, Inc., Copyright ©1984-1996.

FCC CLASS A NOTICE

WARNING: Changes or modifications to this unit not expressly approved by the party responsible for compliance could void this user's authority to operate this equipment.

NOTE: The PowerHub 7000 and PowerHub 8000 have been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15, FCC Rules. These limits are designed to provide reasonable protection against harmfull interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

DOC CLASS A NOTICE

This digital apparatus does not exceed Class A limits for radio noise emission for a digital device as set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class A prescrites dans le reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

VCCI CLASS 1 NOTICE

　この装置は、第一種情報処理装置（商工業地域において使用されるべき情報処理装置）で商工業地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会(VCCI)基準に適合しております。

　従って、住宅地域またはその隣接した地域で使用すると、ラジオ、テレビジョン受信機等に受信障害を与えることがあります。

　取扱説明書に従って正しい取り扱いをして下さい。

This equipment is in the Class 1 category (Information Technology Equipment to be used in commercial and/or industrial areas) and conforms to the standards set by the Voluntary Control Council For Interference by Information Technology Equipment aimed at preventing radio interference in commercial and/or industrial areas.Consequently, when used in a residential area or in an adjacent area thereto, radio interference may be caused to radios and TV receivers, etc. Read the instructions for correct handling.

CE *NOTICE*

Marking by the symbol **CE** indicates compliance of this system to the EMC (Electromagnetic Compatibility) directive of the European Community and compliance to the Low Voltage (Safety) Directive. Such marking is indicative that this system meets or exceeds the following technical standards:

- EN 55022 - “Limits and Methods of Measurement of Radio Interference Characteristics of Information Technology Equipment.”
- EN 50082-1 - “Electromagnetic compatibility - Generic immunity standard Part 1: Residential, commercial, and light industry.”

SAFETY CERTIFICATIONS

ETL certified to meet Information Technology Equipment safety standards UL 1950 3rd Edition, CSA22.2, No. 950-95, EN 60950 (1992) and IEC 950, 2nd Edition.

CANADIAN IC CS-03 COMPLIANCE STATEMENT

NOTICE: The Industry Canada label identifies certified equipment. This certification means that the equipment meets certain telecommunications network protective, operational and safety requirements. The Industry Canada label does not guarantee the equipment will operate to the user's satisfaction.

Before installing this equipment, users should ensure that it is permissible to be connected to the facilities of the local telecommunications company. The equipment must also be installed using an acceptable method of connection. In some cases, the company's inside wiring associated with a single line individual service may be extended by means of a certified connector assembly (telephone extension cord). The customer should be aware that compliance with the above conditions may not prevent degradation of service in some situations.

Repairs to certified equipment should be made by an authorized Canadian maintenance facility designated by the supplier. Any repairs or alterations made by the user to this equipment, or equipment malfunctions, may give the telecommunications company cause to request the user to disconnect the equipment.

Users should ensure for their own protection that the electrical ground connections of the power utility, telephone lines and internal metallic water pipe system, if present, are connected together. This precaution may be particularly important in rural areas.

Caution: Users should not attempt to make such connections themselves, but should contact the appropriate electric inspection authority, or electrician, as appropriate.

TRADEMARKS

FORE Systems is a registered trademark, and *ForeView* and *PowerHub* are trademarks of FORE Systems, Inc. All other brands or product names are trademarks or registered trademarks of their respective holders.

Table of Contents

CHAPTER 5 Global Commands

CHAPTER 6 More Global Commands

CHAPTER 7 System Commands

CHAPTER 8 Media Subsystem

CHAPTER 9 NVRAM Subsystem

CHAPTER 10 Host Commands

CHAPTER 15 Configuring IP Routing

CHAPTER 16 Configuring IP Multicast

CHAPTER 17 Configuring IP/RIP

CHAPTER 18 Configuring IP/OSPF

CHAPTER 19 Configuring AppleTalk Routing

CHAPTER 20 Configuring IPX Routing

CHAPTER 21 Configuring IPX Translation Bridging

List of Figures

List of Tables

Preface

This manual describes the PowerHub user interface and the software commands used to configure and manage the PowerHub for bridging and routing. To learn about the Asynchronous Transfer Mode (ATM) commands, refer to the *PowerHub ATM Software Reference Manual*. To learn how to create and apply filters to control the traffic received and forwarded by the PowerHub, refer to the *PowerHub Filters Reference Manual*.

Chapter Summaries

Chapter 1 - Features Overview - Describes the PowerHub features.

Chapter 2 - Software Subsystems - Describes the Packet Engine boot PROM commands and the subsystems in the PowerHub.

Chapter 3- PowerHub Files - Describes the files that are shipped with the PowerHub. Also describes files that the PowerHub switch itself can create.

Chapter 4 - Using the Command-Line Interface - Describes how to use both the user interfaces.

Chapter 5 - Global Commands - Describes commands for controlling your session environment and saving those commands in a file for use in other sessions.

Chapter 6 - More Global Commands - Describes commands for controlling your session environment and saving those commands in a file for use in other sessions.

Chapter 7 - System Commands - Describes the system subsystem commands that control various system-level settings on the PowerHub.

Chapter 8 - NVRAM Subsystem - Describes the commands used to make changes in the NVRAM subsystem and booting parameters.

Chapter 9 - Media Subsystem - Describes how the PowerHub relates to the physical media and bridging configuration information.

Chapter 10 - Host Commands - Describes commands for modifying the TELNET and TCP configuration used by the PowerHub when you use a TELNET connection to configure and manage the PowerHub.

Chapter 11 - FDDI Commands - Describes the commands used to display, configure, and adjust parameters related to the FDDI connections.

Chapter 12 - TFTP Commands - Describes commands for performing file transfers between the PowerHub and TFPT servers.

Chapter 13 - Bridge Commands - Describes commands for customizing the PowerHub bridge configuration.

Chapter 14 - SNMP Commands - Describes commands for customizing SNMP communities.

Chapter 15 - Configuring IP Routing - Describes commands for configuring the PowerHub as an IP router.

Chapter 16 - Configuring IP Multicast - Describes commands for configuring the PowerHub for IP Multicast routing.

Chapter 17 - Configuring IP/RIP - Describes commands for configuring the PowerHub to exchange IP route information using RIP (Routing Information Protocol).

Chapter 18 - Configuring IP/OSPF - Describes commands for configuring the PowerHub to exchange IP route information using Open Shortest Path First (OSPF).

Chapter 19 - Configuring AppleTalk Routing - Describes commands for configuring the PowerHub as an AppleTalk router.

Chapter 20 - Configuring IPX Routing - Describes commands for configuring the PowerHub as an IPX router.

Chapter 21 - Configuring IPX Translation Bridging - Describes commands for configuring the PowerHub for IPX encapsulation bridging.

Chapter 22 - Configuring DECnet Routing - Describes commands for configuring the PowerHub as a DECnet router.

Appendix A - Lists command to command conversions from the PowerHub Old User Interface to the New User Interface.

Appendix B - Lists the factory defaults for all the PowerHub configuration parameters.

Appendix C - Shows and describes the encapsulation formats of the packet types routed by the PowerHub switch.

Appendix D - Provides reference information about how the PowerHub switch performs and is configured for netbooting.

Appendix E - Provides a pointer to RFC 1340, the “Well-known Ports” RFC.

Related Publications

The following publications are referred to throughout this manual and comprise the PowerHub Reference manual set.

- PowerHub Hardware Reference Manual, MANU0166-01, November 7, 1997
- PowerHub ATM Software Reference Manual, MANU0271-01, November 7, 1997
- PowerHub Filters Reference Manual, MANU0168-01, November 7, 1997

Technical Support

In the U.S.A., you can contact FORE Systems' Technical Support using any one of the following methods:

1. If you have access to the Internet, you may contact FORE Systems' Technical Support via e-mail at:

   support@fore.com

2. You may FAX your questions to "support" at:

   412-742-7900

3. You may send questions, via U.S. Mail, to:

   FORE Systems, Inc.
   1000 FORE Drive
   Warrendale, PA 15086-7502

4. You may telephone your questions to "support" at:

   800-671-FORE or 412-635-3700

Technical support for non-U.S.A. customers should be handled through your local distributor.

No matter which method is used for support, please be prepared to provide your support contract ID number, the serial number(s) of the product(s), and as much information as possible describing your problem / question.

Typographical Styles

Throughout this manual, all specific commands meant to be entered by the user appear on a separate line in bold typeface. In addition, use of the `Enter` or `Return` key is represented as `<ENTER>`. The following example demonstrates this convention:

```
cd /usr <ENTER>
```

File names that appear within the text of this manual are represented in the following style: "...the `fore_install` program installs this distribution."

Command names that appear within the text of this manual are represented in the following style: "...using the **`flush-cache`** command clears the bridge cache."

Subsystem names that appear within the text of this manual are represented in the following style: "...to access the **`bridge`** subsystem..."

Parameter names that appear within the text of this manual are represented in the following style: "...using *`<seg-list>`* allows you to specify the segments for which you want to display the specified bridge statistics."

Any messages that appear on the screen during software installation and network interface administration are shown in `Courier` font to distinguish them from the rest of the text as follows:

```
.... Are all four conditions true?
```

Important Information Indicators

To call your attention to safety and otherwise important information that must be reviewed to ensure correct and complete installation, as well as to avoid damage to the FORE Systems product or to your system, FORE Systems utilizes the following *WARNING/CAUTION/NOTE* indicators.

WARNING statements contain information that is critical to the safety of the operator and/or the system. Do not proceed beyond a ***WARNING*** statement until the indicated conditions are fully understood or met. This information could prevent serious injury to the operator, damage to the FORE Systems product, the system, or currently loaded software, and is indicated as follows:

WARNING!

Hazardous voltages are present. To reduce the risk of electrical shock and danger to personal health, follow the instructions carefully.

CAUTION statements contain information that is important for proper installation/operation. Compliance with **CAUTION** statements can prevent possible equipment damage and/or loss of data and are indicated as follows:

CAUTION

You risk damaging your equipment and/or software if you do not follow these instructions.

NOTE statements contain information that has been found important enough to be called to the special attention of the operator and is set off from the text as follows:

NOTE

If you change the value of the LECS control parameters while the LECS process is running, the new values do not take effect until the LECS process is stopped, and then restarted.

Laser Notice

Class 1 Laser Product:
This product conforms to applicable requirements of 21 CFR 1040 at the date of manufacture.

Class 1 lasers are defined as products which do not permit human access to laser radiation in excess of the accessible limits of Class 1 for applicable wavelengths and durations. These lasers are safe under reasonably foreseeable conditions of operation.

NOTE

The Laser Notice section applies only to products or components containing Class 1 lasers.

Safety Precautions

For your protection, observe the following safety precautions when setting up equipment:

- Follow all warnings and instructions marked on the equipment.
- Ensure that the voltage and frequency of your power source matches the voltage and frequency inscribed on the equipment’s electrical rating label.
- Never push objects of any kind through openings in the equipment. Dangerous voltages may be present. Conductive foreign objects could produce a short circuit that could cause fire, electric shock, or damage to your equipment.

Modifications to Equipment

Do not make mechanical or electrical modifications to the equipment. FORE Systems, Inc., is not responsible for regulatory compliance of a modified FORE product.

Placement of a FORE Systems Product

CAUTION

To ensure reliable operation of your FORE Systems product and to protect it from overheating, openings in the equipment must not be blocked or covered. A FORE Systems product should never be placed near a radiator or heat register.

Power Cord Connection

WARNING!

FORE Systems products are designed to work with single-phase power systems having a grounded neutral conductor. To reduce the risk of electrical shock, do not plug FORE Systems products into any other type of power system. Contact your facilities manager or a qualified electrician if you are not sure what type of power is supplied to your building.

WARNING!

Your FORE Systems product is shipped with a grounding type (3-wire) power cord. To reduce the risk of electric shock, always plug the cord into a grounded power outlet.

CHAPTER 1 Features Overview

This chapter provides an overview of the major features of the PowerHub. The features discussed include:

- Intelligent Packet Switching
- Software
- Network Management

1.1 Intelligent Packet Switching

In the PowerHub 7000 and 8000, much of the packet switching is performed by the Packet Engine (PE). The PE is the centralized packet processing and forwarding engine of the PowerHub. When a packet is received on a segment attached to the PowerHub, the packet is forwarded to the Packet Engine and placed in Shared Memory, where it is examined and either dropped or forwarded, as applicable. The PowerHub 7000 utilizes the first generation Packet Engine, called the Packet Engine 1 (PE1), while the PowerHub 8000 contains the second generation Packet Engine 2 (PE2).

1.1.1 Packet Engines (PE1 and PE2)

The processors in the PE1 and PE2 contain the bridging and routing engines that intelligently examine packet headers for bridging and routing, and modifying them as required for routing. When a non-intelligent Network Interface Module (NIM) receives a packet from one of its ports, it places the packet on the Packet Channel and transfers it directly to the shared packet-buffer memory on the respective PE.

The Main CPU (MCPU) in the PE examines the source and destination addresses in the packet to determine the segments to which the packet needs to be forwarded and the modifications, if any, to be made to the packet. After the necessary modifications are performed, the Input Output/Processor (IOP) queues the packet for transmission on the appropriate destination port(s).

The Packet Engine is also responsible for maintaining complete routing and bridging tables. Caches of route and bridge tables are distributed to intelligent NIMs which make forwarding decisions locally and use the IOPs to queue the packets to the appropriate NIM.

1.1.1.1 Packet Engine 1 (PE1)

The major features of the PE1 are:

- Supports all currently supported NIMs.
- Contains three 100/150MHz RISC (64bit internal-32bit external) processors, each with specialized functions: one MCPU and two IOPs. Installing a packet Accelerator adds another MCPU, increasing the number of processors to four, similar to the PE2 (refer to Section 1.2.1).
- Supports the two 800Mbps packet channels of the Packet-Channel Backplane found in the PowerHub 7000 for a peak bandwidth of 1.6 Gbps. These high-speed channels are implemented and controlled through the incorporation of ten proprietary ASIC devices.

1.1.1.2 Packet Engine 2 (PE2)

The major features of the PE2 are:

- The PE2 is backwards compatible with the PE1.
- Supports all currently supported NIMs.
- Contains four 100/150MHz RISC (64bit internal-32bit external) processors, each with specialized functions: two MCPUs and two IOPs.
- Supports the four 800Mbps packet channels of the Packet-Channel Backplane found in the PowerHub 8000 for a peak bandwidth of 3.2 Gbps. These high-speed channels are implemented and controlled through the incorporation of ten proprietary ASIC devices.

1.1.2 Network Interface Modules (NIMs)

The PowerHub 7000 and 8000 support various interfaces through the use of Network Interface Modules (NIMs). Some NIMs are termed as intelligent NIMs (INIMs). The following paragrah lists the INIMs. Followed that are general descriptions of all NIMs supported by the PowerHub 7000 and 8000, grouped by interface type. For detailed descriptions of all NIMs, refer to the *PowerHub Hardware Reference Manual*.

1.1.2.1 Intelligent Modules

The following NIMs are termed Intelligent NIMs (INIMs). INIMs have the ability to make packet handling and forwarding decisions. These INIMs contain processor and intelligence (firmware) that can relieve the respective PE some of the workload of handling packets. These INIMs forward packets directly to ports that are physically

- PowerCell 700 ATM module
- Single, Dual, Universal Single and Universal Dual FDDI modules
- 6x1 Fast Ethernet (6x1FE) module
- 2x8 Fast Ethernet (2x8FE) module

1.1.2.2 ATM Modules

The PowerCell 700 supports up to two ATM Media Adapters (AMAs). These AMAs support various physical (PHY) ATM interfaces. The interfaces available include, OC-3 Single-Mode, OC-3 Multimode and OC-3 UTP. If two AMAs are installed, one is a primary port, while the other serves as a backup port.

1.1.2.3 FDDI Modules

The FDDI modules are available in both Single and Dual configurations. Each configuration is available with multi-mode MIC, single-mode ST and UTP connectors. Universal Single and Dual modules, with the same adapters types, are also available. Additionally, there are 1x6 and 1x16 FDDI Concentrator modules. These are available with multi-mode mini MICs or UTP connectors.

1.1.2.4 Ethernet Modules

Ethernet modules are available in the following varieties:

6x1 Universal Ethernet Module (UEM)
Provides six slots for installation of Ethernet Media Adapters (EMAs). Any combination of the following EMA types can be installed on the UEM: AUI (10BAse-5), 10Base-FL (FOIRL-compatible), 1-Base-FB, BNC (10Base-2), MAU (Media Access Unit), 10Base-T (UTP).

10x1 10Base-FL
Provides 10 independent 10Base-FL segments; connection for each segment is provided by multimode ST connectors.

13x1
Provides twelve 10Base-T connectors and one slot for installation of a Fast Ethernet Media Adapter (FEMA); the FEMA types are the same as those for the 6x1FE module.

16x1 Provides 16 independent 10Base-T segments. Connections for each segment is provided by an RJ-45 connector.

4x4 Microsegment Repeater Provides four independent 10Base-T segments. Each segment is further divided into four ports and connection to each port is provided by an RJ-45 connector.

4x6 Microsegment Repeater Provides four independent 10Base-T segments. Each segment is further divided into six ports and connection to each port is provided in a 50-pin Champ-style connector.

1.1.2.5 Fast Ethernet (FE) Modules

Fast Ethernet (FE) modules are available in the following varieties:

6x1FE Provides six FE interfaces in the form of individually installed Fast Ethernet Media Adapters (FEMAs). The FEMAs are available with 100Base-TX, 100Base-FX, or 100Base-T4 connectors.

13x1 Provides twelve 10Base-T connectors and one slot for installation of a FEMA. The FEMA types are the same as those for the 6x1FE.

4x8 Provides

R2x8FE The FE repeater module contains two independent 100 Mb/s segments. Each segment contains eight physical FE repeater ports.

1.2 Software Features

The following software features are supported in both the PowerHub 7000 and 8000. This section describes the features that can be found in the PowerHub software. The focus of this section is on system management, rather than configuration and management of network interfaces. The following subjects are discussed:

- Multiprocessor Otimization
- Boot Sources
- Command Line Interface
- File Management System
- Concurrent Command Line Sessions
- Configuration Files
- Parameter Files
- Automatic Segment State Detection
- Segment Statistics
- Traffic Monitoring
- Virtual Local Area networks (VLANs)
- Bridging and Routing
- Route Protocol Statistics
- Security Filters

1.2.1 Multiprocessor Optimization

Multiprocessor optimization minimizes the latency caused in the normal packet forwarding functions due to the processing of management events. By moving these processing-intensive functions to a separate MCPU, the latency of packets in the fast path can be kept to a minimum.

This feature is dependent on having a PE1 with a Packet Accelerator installed. With the accelerator installed, there are four CPUs available. Without the multiprocessor optimization feature, only three CPUs are used. This feature makes use of the fourth CPU by splitting the functions of the single MCPU.

Multiprocessor optimization moves all of the fast path packet processing to one MCPU and retains the slow path and management functions on the other MCPU. Multiprocessor optimization automatically detects the presence of an Accelerator Card at boot time and operates in the appropriate mode. Without the Accelerator Card, the system uses only one MCPU for all functions.

1.2.2 Boot Sources

With the new PowerHub software architecture you can configure

The PowerHub 7000 can be configured to boot from one, or combination, of three sources: floppy diskette, Flash Memory Module, or a TFTP/BOOTP file server. The PowerHub 8000 has only two boot sources; Compact Flash Card or a TFTP/BOOTP file server. Failure of the primary boot source can be prevented by configuring a boot order in Non-Volatile Random Access Memory (NVRAM).

1.2.3 Command-Line Interface

The PowerHub is managed through a UNIX-like command line user interface. Commands can be issued from a management terminal attached to directly through a TTY connection on the PE or indirectly through an in-band (TELNET) connection. Refer to *Chapter 2, Software Subsystems* for a discussion of the software subsystems available within the PowerHub. Refer to the appropriate section of this manual for discussions of the commands available in each subsystem. Refer to the *PowerHub ATM Software Reference Manual* for discussion of the ATM subsystem related commands.

1.2.4 File-Management System

The PowerHub contains global commands to display, copy, rename, and remove files stored on a floppy diskette or in Flash Memory of the PowerHub 7000 or in the Compact Flash Card of the PowerHub 8000. Checksums can also be calculated for files, display directory and volume information of the floppy diskette, Flash Memory Module or Compact Flash Card. Additionally, the Flash Memory Module or Compact Flash Card can be reformatted, if necessary.

1.2.5 Concurrent Command Line Sessions

Up to four management sessions can be open on the PowerHub at the same time. The primary session is always the session on TTY1, a second TTY session can be opened on TTY2. In addition, up to two TELNET sessions can also be opened, simultaneously.

1.2.6 Configuration Files

Configuration changes affected through software commands can be preserved by saving the changes in a configuration file. Changes saved to the file name **`cfg`**, are automatically applied to following a software reboot, provided the **`cfg`** file is present on the boot source.

1.2.7 Parameter Files

Commands can be issued that modify parameters that control user sessions. These parameters include scroll control, TELNET control characters, commands aliases, and timed commands. Changes to the defaults for these session parameters are lost when the session is closed.

These changes to session parameters can be saved in an environment file. At any time during a user session, an environment file can be read (loaded), reinstating the session parameter changes stored in the environment file.

If an environment file is saved under the name `root.env`, it is automatically loaded whenever the PowerHub is logged into under root status. Likewise, an environment file is saved under the name `monitor.env`, the parameters in that file are automatically loaded when logging on with monitor status or the user level is changed from root to monitor during a session.

1.2.8 Automatic Segment-State Detection

When enabled, Automatic Segment-State Detection automatically senses when a link (or something configured on the link) is “bad” or “down”. When a “bad” or “down” link is detected on a particular port, the state of the segment is reflected in the software’s interface tables. *ForeView* Network Management software allows link types to be enabled or disabled on a particular port. Through *ForeView* the state of the following link types on your PowerHub can be learned:

- AUI
- MAU RPTR
- MAU
- BNC
- BNCT
- 10Base-T
- Fiber
- Unknown

NOTE

To disable automatic segment state detection on a UTP port, rename the configuration file to something other than **`cfg`**, and then reboot the PowerHub.

1.2.9 Segment Statistics

The PowerHub displays access method and protocol statistics related to segment and packet activity. For example, state-change statistics for individual segments can be displayed to see how many times a particular segment has gone up or down since the software was last booted. Statistics related to the protocols are briefly described in Section 1.2.13.

1.2.10 Traffic Monitoring

Port activity can be monitored at regular intervals. For example, statistics of packet activity or packet errors and collisions on a particular port can be monitored and graphed.

1.2.11 Virtual Local Area Networks (VLANs)

A Virtual Local Area Network (VLAN) is a collection of segments that share the same group name or protocol interface address. Layer-2 VLANs are created by creating a bridge group. The software comes with a default bridge group called **`default`** that contains all installed PowerHub segments.

Layer-3 VLANs can be created by assigning the same IP, IPX, or AppleTalk interface address to multiple segments. When the software determines a packet is to be sent to a Layer-3 VLAN assigned to multiple segments, the software forwards a copy of the packet on each segment. When this happens, from a physical standpoint, a separate packet has been sent out each physical interface; however, from a logical standpoint, the forwarded packet has been forwarded onto its single destination network or subnet, irrespective of how many physical interfaces that network or subnet is configured on.

1.2.12 Bridging and Routing

The `bridge` subsystem contains commands for configuring and managing the PowerHub as an IEEE 802.1d bridge. Up to 32 network (bridge) groups can be defined, each containing any subset of PowerHub segments.

1.2.12.1 Bridge Table and Bridge Cache

The software maintains a bridge table containing the MAC-layer hardware addresses of devices to which the PowerHub is able to bridge packets. The software maintains this table by automatically adding new and deleting unused entries. In addition, individual entries can be added or removed, including entries that support multi-homed hosts.

Following is an example of a bridge table. Although only a handful of bridge entries are shown in this example, the bridge table usually contains many entries.

```
Bridging table (aging time = 60 minutes)
Ethernet-address   Seg   Rule  Flags
00-00-00-00-00-00  01    none  aged
00-00-c0-ea-9f-17  01    none
08-00-20-10-19-ac  08    none
00-00-c0-ed-61-4a  01    none
08-00-20-0c-5a-48  08    none
02-cf-1f-90-40-23  01    none
08-00-20-0c-3a-a2  02    none
08-00-20-0c-5a-d2  08    none  aged
ff-ff-ff-ff-ff-ff  --1   none  permanent  bmcast
```

In addition to the bridge table, the software maintains a *bridge cache* of the most recently used source-destination pairs. A *source-destination pair* contains a packet's source and destination MAC-addresses. The bridge cache provides a fast path for the bridging software and gives an at-a-glance view of current bridging activity. The bridge cache can be displayed to see the source-destination pairs that are frequently used.

1.2.12.2 802.1d

The PowerHub can be used "right out of the box" as an 802.1d Bridge. The designation 802.1d refers to the IEEE committee number that came up with the spec for this type of bridge. For more information regarding 802.1d bridging, refer to RFCs 1493 and 1525.

1.2.12.3 Spanning-Tree

The bridge software includes implementation of the 802.1d Spanning-Tree algorithm. When enabled, the software identifies and "breaks" loops in the network without requiring configuration changes. Commands in the `bridge` subsystem allow fine-tuning of the Spanning-Tree parameters to fit network needs.

1.2.12.4 IPX Translation Bridging

IPX translation bridging allows one or more IPX networks that span FDDI and Ethernet segments using different packet encapsulations to be configured. This type of bridging is different from 802.1d bridging, which bridges packets based on the MAC-layer hardware address of the devices in the network. IPX translation bridging is used only in IPX networks.

1.2.12.5 IP Routing

Commands in the `ip` subsystem allow segments to be configured for IP routing. Using `ip` commands, IP interfaces can be assigned to individual segments. The IP routing software also supports IP VLANs, enabling a single IP subnet that spans multiple segments to be defined. The following subsections describe major features of the `ip` subsystem. Refer to See *Chapter 15, Configuring IP Routing* for more information about these features and the `ip` commands.

1.2.12.5.1 Routing Information Protocol (RIP)

The `ip/rip` subsystem commands enable the PowerHub to perform IP routing. Using commands in this subsystem, RIP parameters such as talk and listen can be configured on a segment-by-segment basis. Statistics for RIP packets can also be displayed.

1.2.12.5.2 Open Shortest Path First (OSPF)

The `ip/ospf` subsystem contains commands that can be used to configure the PowerHub as an Open Shortest Path First (OSPF) router. OSPF is a routing protocol that enables each participating router to use a topological map of the network to route packets. OSPF routers exchange route information using link state advertisements (LSAs). An LSA is a packet that reports the link states (up or down) of a router's interfaces that are attached to devices in the OSPF network.

1.2.12.6 AppleTalk Routing

The `atalk` subsystem contains commands that can be used to you configure PowerHub segments for AppleTalk Phase-2 routing. AppleTalk zones and interfaces can be defined as well as commands to ping AppleTalk nodes.

1.2.12.7 IPX Routing

The PowerHub can be configured and managed as an IPX router. In addition, the software provides management information about IPX routes and servers through implementation of IPX Routing Information Protocol (RIP) and Service Advertisement Protocol (SAP). RIP or SAP talk and listen parameters can selectively be enabled on a per-segment basis to control the flow of RIP and SAP updates.

1.2.12.8 DECnet Routing

The **dec** subsystem contains commands for configuring the PowerHub to perform DECnet Phase IV routing. Depending upon the configuration of the network, the PowerHub to can be configured to function as a Level-1 router or a Level-2 router. DECnet statistics for the PowerHub (in its capacity as a DECnet node) and for the individual segments configured as DECnet interfaces can also be displayed.

1.2.13 Route Protocol Statistics

The PowerHub can gather statistics for the following Internet routing protocols:

- AppleTalk
- Bridge
- DECnet
- IP
- IPM
- IPX
- OSPFv2
- RIP
- SNMP
- TCP
- TCP/IP

1.2.14 Security Filters

Filters to can be defined and applied to segments or protocol interfaces to control the traffic sent and received on the segments or interfaces. The following types of filters can be defined:

- Bridge filters
- Host (TCP) filters
- IP filters
- IP route filters (RIP and OSPF)
- AppleTalk filters
- IPX RIP and SAP filters

1.3 Network Management Features

The PowerHub has a rich management environment providing comprehensive support for Simple Network Management Protocol (SNMP), as well as local RS-232 and Telnet console support. *ForeView* graphical network management software provides true point-and-click device configuration and runs on a variety of popular management stations.

1.3.1 Network Management System (NMS)

The Network Management System (NMS) manages the PowerHub by sending requests to a software module, or agent, to change the value of one or more variables on the device. For example, an agent reports data such as the number of packets sent, received or dropped on that device. Then, the managed device and the NMS use SNMP, as the common protocol language, to exchange the information requested by the NMS.

1.3.2 Management Information Base (MIB) Agents

Management Information Base (MIB) agents contain definitions of all resources (represented by managed objects within the MIB that are managed by a network management system (NMS). The managed object has properties that hold values, such as routing table information, error counters, and so on.

1.3.3 *ForeView*

ForeView is a graphical-based management application providing a simplified tool for managing the PowerHub. With a point-and-click interface, *ForeView* provides access to PowerHub functions, both system-level and segment-level. *ForeView* can control the PowerHub, monitor errors, control bridge and routing configuration parameters, and display, print, and save statistics.

ForeView integrates the PowerHub system, bridge, and router features into a single application with access and control of all information from one location. It also contains fault management features to troubleshoot, analyze, and monitor multiple Ethernet or FDDI segments using a single network analyzer.

PowerHub statistics are shown in graphical formats, and the physical attributes, such as model and segment type, are displayed on the front panel of a graphical representation of the PowerHub. The graphical representation is displayed when *ForeView* is started. For more information about the *ForeView* Network Management application, refer to the *ForeView Network Management User’s Manual*.

CHAPTER 2 Software Subsystems

This chapter describes the subsystems found in the user interface. Additionally the Packet Engine boot PROM commands are described. This following sections:

- Packet Engine Boot PROM
- User Interface

2.1 Boot PROM Commands

Boot PROM commands are available to the user when the PowerHub is booted. software commands in the boot PROM can be used to configure such things as specifying the boot source. The boot PROM command prompt is displayed as <**PROM-7PE**> for the PowerHub 7000 and <**PROM-8PE**> for the PowerHub 8000.

Some boot PROM commands are used to configure values in non-volatile RAM (NVRAM) of the Packet Engine (PE). These commands also are available in the `nvram` subsystem. NVRAM settings remain in effect regardless of whether set them from the boot PROM interface or from the `nvram` subsystem. Boot PROM commands can be used to perform the following tasks:

- Boot the PowerHub (**boot|b**).
- Display the MAC address (**ethaddr|ea**).
- Display a directory of files on the floppy diskette or Flash Memory Module, in the PowerHub 7000 or Compact Flash Card of the PowerHub 8000 (**ls|dir**).
- Display the contents of a file contained (**more**).
- Write changes to NVRAM (**nvram**).
- Upload or download files to/from the PowerHub (**zreceive** or **zsend**).

2.1.1 Packet Engine Boot PROM Commands

Table 2.1 lists the commands that can be found in the Boot PROM user interface. No management capability is associated with these commands.Refer to the *PowerHub Hardware Reference Manual* for detailed information on Boot PROM commands.

Table 2.1 - Packet Engine Boot PROM Commands

Command and Description
boot|b [-n] [fd|net] Boots the PowerHub, using the device(s) specified as the boot source. Equivalent to the **system reboot** command.
ethaddr|ea Displays the MAC-layer hardware address. Equivalent to the **system ethaddr|ea** command.
ls|dir [_<file-spec>_ **[**_<file-spec>..._**]]** Displays a directory of the files on either a floppy diskette or in the Flash Memory Module, of a PowerHub 7000, or in the Compact Flash Card of a PowerHub 8000. Equivalent to the **ls|dir** Global command.
more [_-<rows>_**]** _<file-name>_ **[**_<file-name>..._**]** Displays a file located on either a floppy diskette or in the Flash Memory Module, of a PowerHub 7000, or in the Compact Flash Card of a PowerHub 8000. Similar to the **type|cat** Global command.
nvram [set|unset|show] Sets, removes or displays settings NVRAM. Equivalent to commands in the **nvram** subsystem.
zreceive|zr|rz [-+27abcehtw] [_<file-name>_**]** Uploads file from a PC or Macintosh that supports ZMODEM or XMODEM onto a floppy diskette (Only implemented in the PowerHub 7000).
zsend|zs|sz [-+27abehkLlNnoptwXYy] _<file-name>_ Downloads a file from the floppy diskette onto a PC or Macintosh that supports ZMODEM or YMODEM (Only implemented in the PowerHub 7000).

2.2 PowerHub Software

PowerHub software version PH_FT4.0.0 supports many new features not available in version 7-2.6.x. Software version PH_FT4.0.0 offers:

- Standardized syntax of commands across all subsystems with improved filtering.
- Logical subsystem grouping of major protocols with additional subsystem groups combining all networking and non-networking aspects of the PowerHub.
- Extensive on-line help with cross referencing of command syntax.
- Specific order of segments with display restrictors to elicit specific segment and port activity.

2.2.1 User Interface Subsystems

The commands to exercise PowerHub features are grouped into subsystems. Each subsystem contains commands that pertain to a particular aspect of PowerHub configuration or management. To display a list of available subsystems, issue the **`subsystems|ss`** command. Issuing **`help`** or **`global help`** command to display a complete list of commands. The following subsystems are supported in the PowerHub user interface.

Table 2.2 - New User Interface Subsystems

Subsystem	**Description**	**For command descriptions, see...**
`atalk`	AppleTalk commands.	*Chapter 19, Configuring AppleTalk Routing*
`bridge`	Bridging, Spanning-Tree, and IPX translation bridging commands.	*Chapter 13, Bridge Commands*
`dec`	DECnet commands.	*Chapter 22, Configuring DECnet Routing*
`fddi`	FDDI commands.	*Chapter 11, FDDI Commands*
`global`	System-wide commands.	*Chapter 5, Global Commands*

Table 2.2 - New User Interface Subsystems

Subsystem	Description	For command descriptions, see...
`host`	Define and display TELNET control characters, display active TCP connections, and display UDP agents.	*Chapter 10, Host Commands*
`ip`	Configure segments for IP.	*Chapter 15, Configuring IP Routing*
`ip/rip`	Commands for configuring IP / RIP.	*Chapter 17, Configuring IP/RIP*
`ip/ospf`	IP / OSPF commands.	*Chapter 18, Configuring IP/OSPF*
`ip/mcast`	IP Multicast commands.	*Chapter 16, Configuring IP Multicast*
`ipx`	IPX routing commands.	*Chapter 20, Configuring IPX Routing*
`ipx/rip`	Commands for configuring the PowerHub as an IPX router.	
`ipx/sap`	Commands for configuring the PowerHub as an IPX router.	
`media`	Define information about the PowerHub's physical links.	*Chapter 8, Media Subsystem*
`nvram`	Configure settings for boot sources, netbooting, and other configuration items in the NVRAM.	*Chapter 9, NVRAM Subsystem*
`snmp`	Define SNMP communities and managers and display SNMP statistics.	*Chapter 14, SNMP Commands*
`system`	Display and manage hardware configuration items, manage files on floppy disk, Flash Memory Module or Compact Flash Card and save and load configuration files. Default subsystem when starting up the PowerHub.	*Chapter 7, System Commands*
`tftp`	TFTP commands.	*Chapter 12, TFTP Commands*

CHAPTER 3 PowerHub Files

This chapter describes the different types of software used by the PowerHub and describes the files shipped with the PowerHub. If it is necessary to upgrade software, contact FORE Systems TAC.

3.1 Types

The use the following types of software:

Packet Engine boot PROM Contains software used by the Packet Engine when it is booted. From this PROM, configuration values—including the boot source—can be changed and stored in NVRAM. The boot PROM prompt is dislayed as:

`<PROM-7PE>, for the PowerHub 7000`
or `<PROM-8PE>, for the PowerHub 8000`

System software Sometimes called "runtime software." Contains most of the commands and features documented in this manual. The runtime software is accessed from the runtime command prompt. The default runtime prompt is dislayed as:

```
1:PowerHub:
```

NIM Boot PROM On intelligent NIMs (such as FDDI, 6x1FE, and PowerCell ATM modules), the boot PROM contains software used by the module when it is booted. The NIM Boot PROM cannot be interacted with directly.

Runtime PROM Contains runtime features used by intelligent NIMs. The runtime PROM software for each intelligent NIM is stored in firmware on the NIMs themselves.

3.1.1 Software Files

The PowerHub software is shipped already installed. All required software and firmware is installed on the Packet Engine and all instlled NIMs. The following sections describe the software and firmware that may be installed on the PowerHub. Software/firmware actually installed depends on PowerHub model and installed NIMs.

3.1.2 PowerHub 8000

`8PE` System software image file; this file resides on the boot source and is loaded when the system is booted.

`ppu-8PE` Packet-Engine boot PROM image file; resident on the boot PROM.

`8atm` Runtime PROM image for PowerCell 700.

`8fddi` Runtime PROM image for FDDI modules; note that FDDI Concentrator modules do not have runtime PROMs.

`8feth` Runtime PROM image for 6x1FE module.

An instance of the appropriate file resides in a PROM on the intelligent module itself. The software is automatically booted.

3.1.3 PowerHub 7000

`7PE` System software image file; this file resides on the boot source and gets loaded when you boot up the system.

`ppu-7PE` Packet-Engine boot PROM image file; this file is resident on the boot PROM.

`7atm` Runtime PROM image for PowerCell 700.

`7fddi` Runtime PROM image for FDDI modules; note that FDDI Concentrator modules do not have runtime PROMs.

`7feth` Runtime PROM image for 6x1FE module.

An instance of the appropriate file resides in a PROM on the intelligent module itself. The software is automatically booted.

3.1.4 Other Files Supplied with the PowerHub

Additional files that may be installed on the PowerHub are files that are used for testing the system.

`extloop` External loopback script; sets the PowerHub up to run an external loopback test, which tests the circuitry on the Packet Engine or a NIM *and* the segment connection hardware.

NOTE FORE Systems recommends running the extloop test before installation. If loopback cables are not available for running the extloop test, run the intloop test.

`intloop` Internal loopback script; sets the PowerHub up to run an internal loopback test, which tests the circuitry on the Packet Engine or a NIM.

`bootdef` Used by the system when the software is booted to identify the name of the system software image and configuration file.

`dispcfg` Configuration file that runs a series of commands that display system configuration information and statistics; this file is useful for diagnosing configuration problems.

3.1.5 PowerHub Created Files

In addition to the files shipped on the system, the following types of files can be created and saved during operation:

`cfg` PowerHub configuration file; created when issuing the **`system savecfg cfg`** or **`tftp svcfg cfg`** command. The configuration file can be saved under any DOS-compatible filename, but the configuration must be manually loaded, unless the user also edits the `bootdef` file to contain the configuration file name.

`root.env` Environment file for root sessions; created when issuing the **saveenv root.env** command. Environment files under any other DOS-compatible filename, but if must be manually loaded.

`monitor.env` Environment file for monitor sessions; created when issuing the **saveenv monitor.env** command. Environment files under any other DOS-compatible filename, but if must be manually loaded.

`powerhub.dmp` Dump file; created if the PowerHub system software exeriences a system crash.

`iop1.dmp` Another type of dump file the software can produce when a crash is exerienced.

iop2.dmp Another type of dump file the software can produce when a crash is exerienced.

NOTE The presence and contents of any dump (.dmp) files should be supplied to FORE Systems TAC when reporting system crashes. The contents of these dump files can assist TAC in determing the cause of the crash.

CHAPTER 4 Command-Line Interface

This chapter describes how to use the command-line interface. The following sections are discussed:

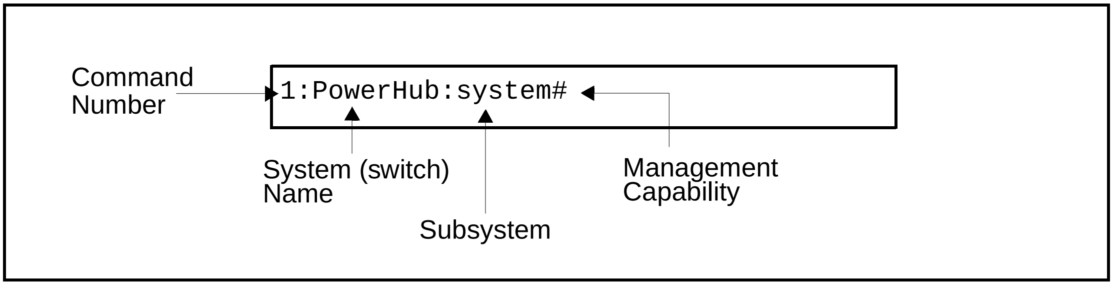
- The user interface
- Command syntax
- Displaying on-line help

4.1 Using the User Interface (UI)

The user interface (UI) comes up by default is the **`system`** subsystem when the PowerHub is initially boot. The following sections describe the items in the runtime prompt and how to issue commands.

4.1.1 Runtime Prompt

Regardless of whether accessing the PowerHub through a TTY (RS-232) port or through a TELNET session, the command prompt is displayed such as the one shown in Figure 4.1:



```
1:PowerHub:system#
```

Figure 4.1 - New User Interface Command Line

As shown in Figure 4.1, the command prompt has four components:

Command Number A sequential number of the commands executed in this session (similar to a command number in the UNIX C-shell). In this example, the command number is `1`.

When a carriage return (`Enter` key) is issued, the PowerHub attempts to execute the command entered at the command prompt. A message or data (if requested) is displayed, then a new command prompt is displayed. The number in the command prompt is one number higher than the number in the previous command prompt.

System Name The name assigned to this PowerHub. The default name, PowerHub, can be changed using the **`system sysname`** command.

Subsystem The name of the subsystem currently in use. Commands issued at the command prompt must either be global commands or commands within the current subsystem. In this example, the subsystem is `system`, the initial subsystem.

Management Capability Indicates whether the session is in *monitor* or *root* capability:

> Indicates monitor (display only) capability. *Monitor* capability statistics configuration information to be dislayed. Commands that change the configuration, clear statistics, or modify internal tables are not allowed.

Indicates root (configuration) capability. *Root* capability allowsany command, including commands that change the configuration and the internal tables to be issued.

In this example, management capability is shown as #, for root capability.

NOTE If a user session is started and the `login`: prompt is displayed, rather than the command prompt, a password must be entered before being allowed to proceed.

The command prompt described in this section allows system software commands to be issued. The `<PROM-7PE>` or `<PROM-8PE>` prompt can be used to issue some commands, including `NVRAM` commands, but does not allow system software commands. Refer to *Chapter 2, Software Subsystems* for information on the `Boot PROM commands`.

4.1.2 Entering and Editing Command Text

All commands are typed at the command prompt using a keyboard attached to the workstation, terminal, or PC being used as a management station. The workstation or terminal must be attached to one of the TTY ports or connected through an active TELNET session.

Commands and arguments are case-sensitive and should be entered only as shown in the manual or on-line help. Each command must fit on a single line and cannot exceed 128 characters in length. The keys used to edit and issue commands are the standard keys used on most UNIX workstations:

- To issue a command, enter the command name and arguments (if needed) after the command prompt, then press `<Enter>`.
- To erase individual characters in a command, use the `<Backspace>` or `<Delete>` key, or the `EraseChar` character assigned in the TELNET session (usually `<Ctrl+H>`).
- To cancel an entire line of input, use the reassign character (usually `<Ctrl+U>`).
- To control the scrolling of output on the terminal, use `<Ctrl+S>` to stop the flow and `<Ctrl+Q>` to resume the flow.

Commands in the **`host`** subsystem can be used to display or change the key sequences used in TELNET sessions. The key sequences for the current session or the default key sequences used for all sessions can also be dislayed or changed.

4.2 Command Syntax

This section describes and provides examples of the command syntax used in the UI. The following subsections describe the syntactical elements of the UI:

4.2.1 Verb Objects

This following sections describe the objects used in the UI.

4.2.1.1 set and unset

These verbs set or remove the setting from system parameters. Examples include:

- The boot order (the order in which the PowerHub attempts to use the floppy diskette, Flash Memory Module, Compact Flash Card, or TFTP boot server for booting the software).
- Scroll control (stty) parameters
- Timed commands
- Routing protocols
- Specific bridging and routing protocol features

When the **`set`** or **`unset`** verb is prepended by **`c`** or **`p`**, or **`n`** or **`s`**, the verb applies only to specific segments (**`c`**,**`p`**) or specific networks (**`n`**,**`s`**).

4.2.1.2 define and undefine

These verbs create or delete templates and rules, both of which are components of filters.

4.2.1.3 attach and detach

These verbs apply or remove filters (created using the **`define`** and **`undefine`** verbs) to segments or protocol interfaces.

4.2.1.4 add and delete

These verbs add or delete objects from tables. Examples include bridge-table entries, protocol interface-table entries (IP, AppleTalk, IPX, and DECnet), and route-table entries.

4.2.1.5 enable and disable

These verbs turn on or off specific software features. Examples include bridging and protocol routing, specific IP routes, IP Helper, and so on. When the **`enable`** or **`disable`** verb is prepended by **`p`**, or **`n`** or **`s`**, the verb applies only to specific segments (**`p`**,**`s`**) or specific networks (**`n,s`**).

4.2.1.6 show and clear

These verbs display and clear (if applicable), configuration information, tables, caches, and statistics. With these verbs configuration information can be dislayed or statisticscan be dislayed or cleared.

4.2.2 Noun Objects

This section describes the nouns in the UI in the following subsections.

4.2.2.1 config

Shows the parameter settings for the hardware, the bridging subsystem, and the routing protocol subsystems. In general, the **`config`** (or **`show config`**) command displays system parameters that can be configured through the software.

4.2.2.2 status

Shows the current status of the hardware (segment up / down status, 10Base-T port status, and so on), the current bridge status of segments (bridging enabled or disabled, Spanning Tree enabled or disabled), and so on. In general, the configuration parameters are displayed when the **`status`** (or **`show status`**) command is issued.

4.2.2.3 stats

When the **`stats`**, or **`show stats`**, command is issued, statistics related to the feature area in the current subsystem (or specified subsystem, if different from the current) are displayed. For example, the **`stats`** command issued from within the **`ip`** subsystem displays IP, ARP, and ICMP packet statistics.

4.2.2.4 interface

A routing protocol interface (IP, IP Multicast, AppleTalk, or IPX).

4.2.2.5 route

An IP, IP Multicast, AppleTalk, IPX, or DECnet route.

4.2.2.6 bt

The bridge table.

4.2.2.7 cache

Contains “fast-path” entries. The bridge subsystem, and all routing protocol subsystems, contain a cache. The “fast-path” entry is a shortcut used to bridge or route packets. When a bridge table or route table is in the fast path, the PowerHub does not need to perform all the bridging or routing processing that it normally performs in order to bridge or route a packet. The PowerHub maintains each cache by placing in them the most recently used source destination MAC-address pairs (for bridging) or protocol interface addresses (for routing).

4.2.3 Parameters

This section describes the types of parameters in the UI.

4.2.3.1 Keyword Parameters

A keyword parameter is a parameter that can be entered at any point following the verb.

4.2.3.2 Positional Parameters

A positional parameter is a parameter that must be entered in a specific position following the verb of a syntactical command. The need for positional parameters in the UI is infrequent because the software uses keywords to determine the function being performed. When the need for a positional parameters arises, the software provides a resonse with the correct position of parameters in the command.

4.2.4 Online Help

PowerHub software version PH_FT4.0.0 offers an improved version of on-line help over previous PowerHub software versions. The **help|?** command offers three levels of help.

The first level of on-line help gives a command's syntax based on the search of a *verb* command (action) and describes the command as in the following example:

```
[show] dd|default-device
```

The second level of on-line help gives a command's syntax based on the search of a *noun* command (object) and describes the command. In the following example, the **ip** subsystem is used to display a portion of the help available for the **arp** command:

```
arp [show] [-r] [-t] [-s] [<disp-restrictors>]
```

By specifying additional parameters to a command, more specific on-line help is available as in the following example:

```
arp [show] age Show ARP aging time
```

CHAPTER 5 Global Commands

This chapter describes how to use global commands to perform the following tasks:

- Boot the software
- Log in and log out
- Change management capability between root and monitor

NOTE

Additional global commands are discussed in *Chapter 6, More Global Commands*. These commands allow other things to be executed, i.e., display help, list subsystems, etc. For a complete listing of global commands, see Appendix A.

5.1 Rebooting the Switch

The PowerHub can be booted, or rebooted,using any of the following methods:

- Turning the power supplies off, then back on.
- Pressing the reset switch (labeled RST), located on the front of the Packet Engine.
- Issuing the **`boot`** (**b**) command at the boot PROM prompt.
- Issuing the **`system reboot`** command.

5.1.1 Turning the ower Sulies Off/On

Turning the power supplies off and then on causes the PowerHub to reload the system software.

NOTE

If there are multiple power supplies installed in the chassis, turn all power supplies off, and then on, at the same time. Refer to the *PoiwerHub Hardware Reference Manual* for detailed instructions.

5.1.2 Reset Switch

The reset switch is located on the front panel of the Packet Engine and is labeled RST. When the reset switch is pressed, the Packet Engine performs a “cold” restart. During a cold restart, the a power-on self-test is conducted to check various hardware components.

Depending on the boot preference(s) specified, the Packet Engine configures the PowerHub for runtime operation. For an example of the boot messages displayed and additional methods for rebooting, refer to the *PowerHub Hardware Reference Manual*.

5.1.3 Booting the PowerHub Software

The **`boot|b`** command found at the boot PROM prompt can be used to boot the PowerHub. The syntax for this command is:

```
boot|b -n [fd|net|fm]
```

fd Boots the software from floppy diskette (PowerHub 7000) or the Compact Flash Card (PowerHub 8000).

net Boots the PowerHub over the network, if the PowerHub is configured for netbooting.

fm Boots the software from the Flash Memory Module (PowerHub 7000 only).

If a boot source is not secified, the boot order configured in NVRAM is used. If a boot order has not been configured in NVRAM, the floppy drive, PowerHub 7000, or Compact Flash Card, PowerHub 8000, (**`fd`**) is used.

NOTE

The system software image file (7PE) must be present on the boot source specified. The **`boot|b`** command does not affect the boot order specified in NVRAM.

5.1.4 Using the Reboot Command

Issue the **`reboot`** command from the **`system`** subsystem to boot the PowerHub. Following is an example of the boot messages displayed:

```
32:PowerHub:system# reboot
FORE 7000 PE
Prom version: 7pep-2.5.7 (s1.80) 1996.02.13 14:21
I-cache 16K OK
Entering cached code
```

```
I-cache execution OK
D-cache 4K OK
SRAM 128K OK
DRAM ..............................................24064K OK
Shared Memory ...2048K OK
Entering Monitor
FlashInit: found 2MB Flash Memory Module
Board Type: 7PE ,  CpuType: MCPU,  Instance: 2
Ethernet address: 00-00-ef-02-b9-c0
(normal start)
Hit any key now to abort boot [0]:
Trying floppy boot...
Boot definition file: bootdef (default)
Using disk bootdef, parsed as version 0
Loading file "7pe" (AB format) |
File loading complete
initial program counter: 0x80210b40
Disabling break interrupt 4
Switching interrupt source back to ID bits
FORE PowerHub 7000 Runtime System Software
7pe-BRULEE-Alpha33 (s1.445) 1996.12.05 00:45
tty driver initialized
system timer initialized
floppy driver initialized
FlashInit: found 2MB Flash Memory Module
memory manager initialized
Board Type: 7PE ,  CpuType: MCPU,  Instance: 2
System ethernet address: 00-00-ef-02-b9-c0
Looking for packet accelerator card
Did not find packet accelerator - will use 2MB shared memory
Shared mem available: 2024448
PE: slot 5
X Bus:
Slots that are equipped:
Slots that are equipped and latched:
Y Bus:
Slots that are equipped: 2 1
Slots that are equipped and latched: 2 1
```

The boot messages shown in this example are displayed when the software is booted from the floppy drive. The PowerHub can be configured to boot from the Flash Memory Module, Compact Flash Card or from a TFTP file server.

NOTE

If a NIM fails to come up when the system is boot, check to ensure that the left ejector handle is pressing on the activation switch, located behind the ejector handle. If the switch is not being pressed, the NIM does not operate. Refer to the *PowerHub Hardware Reference Manual* for more information on the activation switch

5.2 Logging In and Out

This section describes how to log in to and out of the Powerhub hrough either a direct connection (TTY) or in-band connection (TELNET).

5.2.1 Logging In

When the Lock Switch on the Packet Engine is set to the unlocked position (U), it is not necessary to log in. When the system is boot (or reboot), a command prompt is displayed. However, if the Lock Switch is set to the Locked position (L), a login: prompt is displayed. Enter “root” or “monitor,” then a password, when prompted. The password entered depends upon the management capability desired, or allowed:

- If “monitor” is specified, enter the password for monitor capability.
- If “root” is specified, enter the root password.

When the system is shipped from the factory, the password for each management capability is blank. At the password: prompt, press `<Enter>`. Use the **`system passwd`** command to set or change passwords.

5.2.2 Logging Out

To log out of a PowerHub session, issue one of the following commands:

- **`logout`**
- **`bye`**

These commands end the session from which the command is issued, but do not affect other user sessions. All sessions on the PowerHub can be exited by powering down or rebooting.

Configuration changes to the PowerHub remain in effect until the next software reboot. If changes are saved in a configuration file, they can be reinstated following reboots. To save environment changes, save them to an environment file or the environment changes are lost. Settings to the PowerHub can be reinstated by reading the environment file into a user session.

NOTE

Configuration changes affect only the PowerHub. Environment changes affect the user session from which they are made but do not affect configuration.

CHAPTER 6 More Global Commands

This chapter describes how to perform the following tasks:

- Displaying and using command histories
- Managing files
- Defining and using command aliases
- Defining and using timed commands
- Saving and reading environment files (session settings)

6.1 Displaying and Using Command Histories

For each session, the PowerHub maintains a history of the 32 most recently issued commands. Using the history commands, the command history can be displayed, reissue commands, or edit and reissue commands. To display the 32 most recently issued commands, issue the **`history|hi`** command. To reissue or edit commands listed in the command history, use the *history control characters*. The default history control characters.

! History-prefix character.

^ Quick-substitution character.

The history control characters can be used to form commands to reissue (or modify and reissue) commands from the command history. The history commands used to edit and reissue commands listed in the command history are discussed below. The syntax is shown using the default history characters.

!! Repeats the previous command.

! <n> Repeats a command listed in the command history, where <*n*> indicates the number of the command as listed in the history.

! <-i> Issues a previously issued command, where <*i*> is the offset back from the current command. For example, the command `!-1` gives the same results as `!!`, reissuing the previous command.

! <substring> Repeats a previous command that begins with the string identified by <*substring*>.

^<old>^<new> Modifies, then reissues the previous command, where *<old>* indicates the string to be replaced with *<new>*.

Use the **histchars** command to display the current history control characters. To change the history control characters, issue the **histchars** command with one or both optional arguments:

histchars [*<ch1>***[***<ch2>***]]**

Following is an example of the use of this command:

```
5:PowerHub:system# histchars
history sub: !quick sub: ^
```

6.1.1 Resetting the Return Code

Use the **rcprompt enable** command to reset return codes for commands executed automatically from a script. The syntax for the **rcprompt enable** command is.

rcprompt enable|disable

enable|disable Enables printing of command return codes in the next prompt. Return codes are displayed with **0** for successfully executed commands. This feature is intended primarily for automated interactions with the PowerHub command-line interface.

6.2 Managing Files

This section describes how to manage files stored on the local storage devices (Flash Memory Module, Compact Flash Card and floppy drive).

6.2.1 Displaying a Directory

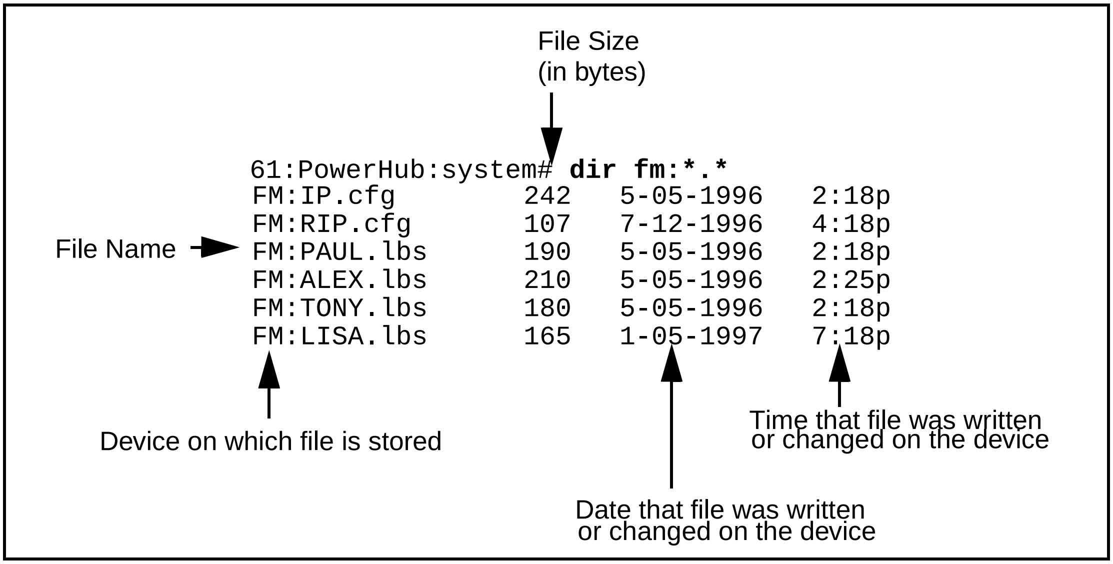
Use the **ls|dir** command to display a directory listing of files stored on the specified device. The syntax for this command is:

```
ls|dir <default-device>|[fd:|fm:] [<filespec>]
```

<default-device>|[fd:|fm:] List the files specified in the <*filespec*> located on either the default-device or the floppy diskette or Compact Flash Card (**fd:**) or in the Flash Memory Module (**fm:**).

<filespec> Specifies a file name. Wildcards (***** and **?**) can be used for any portion of the file name.

This command is designed to present directory information in a way familiar to DOS or UNIX users. Figure 6.1 shows the display produced by the **dir** command against the Flash Memory Module.



```
61:PowerHub:system# dir fm:*.*
FM:IP.cfg         242   5-05-1996   2:18p
FM:RIP.cfg        107   7-12-1996   4:18p
FM:PAUL.lbs       190   5-05-1996   2:18p
FM:ALEX.lbs       210   5-05-1996   2:25p
FM:TONY.lbs       180   5-05-1996   2:18p
FM:LISA.lbs       165   1-05-1997   7:18p
```

File Size
(in bytes)
File Name
Device on which file is stored
Time that file was written or changed on the device
Date that file was written or changed on the device

Figure 6.1 - Details of the **dir** Command Display

6.2.2 Deleting Files

The **rm** command is used to remove (delete) files. The syntax for the **rm** command is:

```
rm [-f] [-i] <default-device>|[fd:|fm:] [<filespec>]
```

-f Forces the software to remove the file(s), without asking you if you are sure before removing each file.

-i Overrides the **-f** (Force) flag, presenting a prompt before removing each file. The prompt gives you the opportunity to cancel your request to remove the file. If you do not specify **-f** or **-i**, **-i** is the default.

<default-device>|[fd:|fm:] Deletes the files specified in the <*filespec*> located on either the default-device or the floppy diskette or Compact Flash Card (**fd:**) or in the Flash Memory Module (**fm:**).

<filespec> Specifies a file name. Wildcards (***** and **?**) can be used for any portion of the file name.

6.3 Command Aliases

The command-line interface provides an *alias* mechanism that allows frequently-used commands to be issued with just a minimum of keystrokes. Each time a command is issued, the alias can be entered instead of the command itself. The alias mechanism is a simplified version of the alias mechanism found in the UNIX C-shell.

Aliases are local to the current command-line session. That is, they are not remembered across logins or resets unless saved to an environment file. Each session can have up to 32 aliases. Aliases can be stored in an environment file by issuing the **system saveenv** <*file-name*> command.

When alias is used from the command line, it must be the first item after the command prompt. However, a subsystem name can be entered before the alias; for example:

```
media box
```

media Is the subsystem name.

box Is the alias.

An alias can be used to make a frequently used command global; for example: **alias box media showcfg**.

In addition to entering an alias directly from the command line, an alias can be used as part of a timed command. When the timed command is activated, the command represented by the alias is issued.

6.3.1 Defining an Alias

To define an alias, issue the following command:

alias *<name> <command>*

<name> The name defining the alias.

<command> The command (including arguments) to assign to the specified alias.

For example, to define“?” as an alias for “help,” type:

```
9:PowerHub:system# alias ? help
Added ?:help
```

The PowerHub acknowledges that it has added **?** to its list of aliases. To define “hist” as an alias for “history,” issue the following command:

```
10:PowerHub:sysyem# alias hist history
```

Note that only one level of alias substitution is performed. That is, strings within an alias definition are not checked against the alias list. For instance, in the following example, the “?” alias for help still works even though “help” is defined as an alias for subsystems.

```
11:PowerHub:system# alias help subsystems
Added help:    subsystems
12:PowerHub:system# help
atalk atm atm/1483encap atm/clip atm/foreip
atm/lane bridge dec fddi host ip ip/rip ip/
ospf ip/mcast ipx ipx/rip ipx/sap media nvram
snmp system tftp
13:PowerHub:system# ?
Global commands:
history|hi     show command
history
...help listing continued
```

6.3.2 Displaying an Alias

To display the definition of an alias, issue the following command:

alias [*<string>***]**

<string> The alias for which to display the definition. If an alias is not specified, all aliases defined for the current session are displayed.

Following is an example of the display produced by this command:

```
14:PowerHub:system# B
?       help
help    subsystems
```

6.3.3 Saving and Loading an Alias

Aliases apply to the current command-line session only. For example, if aliases are defined within a TTY1 session, then a second TTY session or a TELNET session is opened, the aliases are not available to the new session. Moreover, a TTY1 session is ended without saving the aliases defined during that session, they are lost.

There are several ways to save aliases. The easiest way is to save them to an environment file using the following command:

saveenv *<file-name>*

Aliases can be manually added to a file, then type **readenv** *<file-name>* to read (load) them at each log in. Environment files contain other session parameters in addition to aliases.

Alternatively, aliases can be placed in an external file on the management station, then loaded into the command-line session at each log in. For example, if the management station is a PC running Kermit, the “aliases” can be put into a text file on the PC, then loaded into the PowerHub by escaping to the Kermit prompt and typing “transmit aliases”; this command transmits the text file as if it were typed. On UNIX systems running the tip program, use the “~>” escape to send a local text file to the PowerHub.

6.3.4 Deleting an Alias

To delete an alias, issue the following command:

unalias *<name>*

<name> The alias to delete.

Following is an example of the use of this command. In this example, the “?” is a string that was defined as an alias by an **alias** command:

```
15:PowerHub:system# unalias ?
```

6.4 Using Timed Commands

When a command-line session is used to monitor network behavior, certain commands may be executed regularly and repeatedly. For example, it may be necessary to display a bridge or route cache at regular intervals to observe frequently requested bridge or route destinations for certain segments.

Timed commands can be defined to automatically issue any command string at a regular interval. A *timed command* is similar to an alias, but is automatically issued by the PowerHub at a specified interval. PowerHub commands and even aliases can be defined as timed commands. Each user session can have up to eight timed commands. Table 6.1 lists the commands used to define, display, activate, and delete timed commands.

Table 6.1 - Timed Commands

Use Command...	To...
timedcmd | tc	Display all timed commands.
timedcmd | tc add<*id*> <*interval(secs)*> <cmd-and-args>	Add a timed command.
timedcmd | tc enable <*id*>	Start the timer for a timed command.
timedcmd | tc disable <*id*>	Stop the timer for a timed command.
timedcmd | tc del <*id*>	Delete a timed command.

As with command aliases, timed commands are local to the current command-line session. That is, they are not remembered across logins or resets unless saved to an environment file. Also, timed commands are automatically canceled when the command-line session ends.

Timed commands can be stored in an environment file by issuing the **saveenv** *<filename>* command.

6.4.1 Defining a Timed Command

Use the **`timedcmd add`** command to define a timed command. The syntax for this command is:

```
timedcmd|tc add <id> <interval(secs)> <cmd-and-args>
```

<id> Specifies the name of the timed command. The timed command is activated by issuing this name. Specify an alphanumeric string up to 15 characters in length.

<interval (secs)> Specifies, in seconds, the interval at which the timed command is reissued. Specify a minimum of 1 second.

<cmd-and-args> Specifies the command string issued each time the interval specified by *(secs)* expires. Specify a command, including its arguments, or an alias.

NOTE

The subsystem must be included in each timed command. For example, if a timed command is created that issues the **`interface`** command from within the **`atalk`** (AppleTalk) subsystem, specify the command as **`atalk interface.`**

Following is an example of how to define a timed command. In this example, a timed command named “**`bcache`**” is defined to automatically display the bridge cache every ten seconds. A command such as this is useful for quickly observing bridge activity.

```
35:PowerHub:system# timedcmd add bcache 10 bridge display-cache 1-6
Added bcache: 10 secs, bridge display-cache 1-6 (timer not running)
```

6.4.2 Starting a Timed Command

To use a timed command, the timer for the command must be started. When the timer is started, the command is issued when the specified timer value expires. The PowerHub continues to issue the timed command each time the interval expires until the **`timedcmd disable`** command is issued, or until the session is logged out. Note that if the **`system saveenv`** *`<file-name>`* command is issued while the timed command is running, the **`timedcmd enable`** *`<id>`* command is added to the environment file. Consequently, the timed command is started again the next time the environment file specified by *`<file-name>`* is read. Use the **`timedcmd enable`** command to start the timer for a timed command. The syntax for this command is:

```
timedcmd|tc enable <id>
```

<id> Specifies the name of the timed command to start the timer. Make sure the name of the timed command itself is specified, rather than the command string associated with the timed command.

6.4.3 Stopping a Timed Command

Use the **`timedcmd disable`** command to stop a command timer. When a command timer is stopped, the PowerHub stops issuing the timed command. The syntax for the **`timedcmd disable`** command is.

```
timedcmd|tc disable <id>
```

<id> Specifies the name of the timed command. Make sure the name of the timed command itself is specified, rather than the command string associated with the timed command.

A timed command can be stopped by ending the command-line session from which the timed command was started. The timed command can be restarted by issuing the **`timedcmd enable`** *`<id>`* command.

If an environment file is saved while the timed command is running, or manually added the timed command (and the **`timedcmd enable`** *`<id>`* command) to the environment file, the timed command starts again when the environment file is read (loaded).

6.4.4 Deleting a Timed Command

Use the **timedcmd del** command to delete a timed command. The syntax for this command is:

timedcmd|tc del *<id>*

<id> Specifies the name of the timed command to delete.

Following is an example of the use of this command:

```
37:PowerHub:system# timedcmd del bcache
bcache: deleted
```

6.5 Using Environment Files

At any time during a command-line session, an *environment file* can be saved or loaded, Environment files are ASCII files that contains PowerHub commands that can define the following parameters:

- Scroll (stty) parameters (maximum number of rows and “more” enable or disable; defined using the **stty** command).
- Command aliases.
- Timed commands

When a session is ended by logging out or rebooting, changes made to these environment settings are lost. However, if the environment file is saved before logging out or rebooting, environment settings are placed into an environment file.

If the environment file is loaded during a command-line session, the commands in the file recreate the parameters that were active when you saved the file. Following is an example of an environment file:

```
#
# stty
system stty rows 24
system stty more enl
#
# aliases
#
system alias br      bridge s all pi,po
system alias btc     bridge btc
#
# timed commands
#
system timedcmd add bcache 10 bridge display-cache 1-6
```

Notice that the file contains three sections: **stty**, **aliases**, and **timed** commands, shown in bold type in the example. The sections are labeled by comment lines, which begin with #. The **stty** section contains commands to set the "more" feature and specify the maximum number of rows to be displayed when the "more" feature is enabled. The **alias** section contains **alias** commands that define various aliases. In this example, aliases are created to display tables, and ping IP addresses. The **timed** commands section contains a command that defines the timed command **bcache**.

Multiple environment files can be loaded during a command-line session; the effects are cumulative. Note, however, that earlier settings can be overwritten by later settings. For example, if a file that contains the **stty +more** command (to enable the "more" feature) is loaded, then another file that contains the **stty -more** command is loaded, the net effect is that "more" is disabled. If unexpected results are experienced when using multiple environment files, examine the files to ensure that parameters are not being overwritten inadvertently.

An environment file can be automatically loaded at the beginning of a command-line session by saving the file as one of the following:

root.env If the PowerHub is booted, and either the Lock Switch is off or if the log in is under root capability, this file is loaded, if present.

monitor.env If the PowerHub is booted, and the Lock Switch is on, and the log in is under monitor capability, this file is loaded, if present.

NOTE

If an environment file is edited manually, do not place into the file any commands associated with using the floppy drive or the Flash Memory Module. This includes commands such as **readenv** and **readcfg**, which read files.

Also, if the environment file contains the **timedcmd enable** command and the timed command it starts has been defined in the environment file or earlier in the user session, the timed command is started when the environment file is read.

The following sections describe how to save environment settings and how to activate the settings during a user session.

6.5.1 Saving an Environment File

To save an environment file, issue the following command:

saveenv|svenv *<file-name>*

<file-name> Specifies the name of a file that contains the environment parameters. Up to eight alphanumeric characters plus a three-character extension can be specified. Use a period to separate the name from the extension. It is recommended that the extension **env** be used (for example: Lab1.env) to distinguish environment files from other files.

NOTE

If in monitor mode, `root.env` files cannot be saved.

This command assumes that the environment file is written to the default local storage device. The storage device can be explicitly specified by prefacing the file name with **fd:** (floppy drive or Compact Flash Card) or **fm:** (Flash Memory Module).

Following is an example of how to save a default environment file. In this example, the command-line session is in monitor capability. Correspondingly, the current environment settings are saved into a file called `monitor.env`:

```
32:PowerHub:system> saveenv monitor.env
monitor.env: Environment saved
```

In the following example, the scroll parameters are set, then some aliases and a timed command are defined. These environment settings are then saved into an environment file named `Lab1.env`. At any time during a command-line session, this file can be loaded to activate the environment settings it contains.

All current environment settings (aliases, timed commands, and scroll parameters) are saved in the environment file. These commands create the environment file shown in the following example. Environment files can save you a lot of typing:

```
33:PowerHub:system#stty rows 24
34:PowerHub:system# stty -more
35:PowerHub:system# alias aarp    atalk at *.1
36:PowerHub:system# alias br      bridge s all pi,po
37:PowerHub:system# alias bru     bridge s all pu,cu
38:PowerHub:system# alias bsc     bridge sc
39:PowerHub:system# alias bt      bridge bt
40:PowerHub:system# alias btc     bridge btc
41:PowerHub:system# alias clr     system rdcfg clearhub
42:PowerHub:system# alias ethan   ip ping 181.17.45.17
43:PowerHub:system# alias sascha  ip ping 191.1.45.3
44:PowerHub:system# alias bill    ip ping 131.24.45.2
45:PowerHub:system# alias ginger  ip ping 131.24.45.2
46:PowerHub:system# alias sat     atalk s ddp
47:PowerHub:system# alias si      ip s ip
48:PowerHub:system# alias sx      ipx s ipx
49:PowerHub:system# timedcmd      add bcache 10 bridge display-cache 1-6
50:PowerHub:system# saveenv monitor.env
Lab1.env: Environment saved
```

6.5.2 Loading Environment Files

Environment files can be loaded and thereby activate environment settings saved in that file using the following command:

readenv|rdenv *<file-name>*

<file-name> Specifies the name of the file that contains the environment settings. If the file is found, the commands in the file are executed; otherwise, an error message is displayed.

NOTE

This command assumes that the environment file is located on the default local storage device. The storage device can be explicitly specified by prefacing the file name with **fd**: (floppy drive or Compact Flash Card) or **fm**: (Flash Memory Module).

6.5.3 Editing Environment Files

As an alternative to setting environment parameters, then using **saveenv** to create or edit an environment file, the file can be edited directly using a text or ASCII editor.

NOTE

Do not place any commands associated with using the floppy drive, Compact Flash Card or Flash Memory Module in the file. This restriction includes commands such as **readenv** and **readcfg**, which read files.

CHAPTER 7 System Commands

This chapter describes the commands in the system subsystem. The system subsystem commands control various system-level settings on the PowerHub. The following tasks can be performed using these commands:

- Set or display the baud rate on TTY1 and TTY2.
- Display the date and time the PowerHub was last loaded and the boot order.
- Remove and install modules while the PowerHub is operating.
- Display the current configuration
- Set or display the system date and time.
- Enable or disable automatic segment state detection.
- Read the MAC-layer hardware address.
- Display prom information of installed modules.
- Set or change management level passwords.
- Read the default or an alternate configuration file.
- Reboot the PowerHub.
- Save the default or an alternate configuration file.
- Assign a location for the PowerHub.
- Assign a name to the PowerHub.
- Display the temperature of one or all modules installed.
- Enable or disable the TTY2 port.
- Display the elapsed time since the last reboot.
- Display the software version string and idprom info of the cards in the PowerHub.

7.1 Accessing the System Subsystem

The `system` subsystem is the default subsystem entered when the PowerHub completes the boot process. To access the `system` subsystem from any other subsystem, enter **`system`** from the current runtime prompt.

7.2 System Commands

There are several commands in the `system` subsystem that give information on the state and configuration of the PowerHub. These commands allow configuration and environmental information about the PowerHub to be displayed.

7.2.1 Changing the TTY Port Baud Rate

The baud rate (data transmission rate) for the RS-232 ports (tty1 and tty2) can be changed using the **baud** command. However, before setting the baud rate associated with tty2, the tty2 port must be enabled using the **tty2** command (see Section 7.2.15). The syntax for this command is:

```
baud set tty1|tty2 1200|2300|4800|9600|19200
baud [show]
```

set Sets the specified baud rate for the specified port.

tty1|tty2 Specifies the port to be set.

1200|2300|4800|9600|19200 Select one of the listed baud rates to apply to the specified port.

The newly specified rate is stored in NVRAM and takes effect immediately. It is retained across logins and power cycles. An example of the use of this command is:

```
43:PHswitch:system# baud
TTY        Baud Rate
1         9600
2         19200
44:PHswitch:system# tty2 enable
45:PHswitch:system# baud set tty2 9600
Changed tty2 baud rate to 9600; written to nvram
46:PHswitch:system# tty2 disable
tty2 is now closed
47:PHswitch:system#
```

NOTE

If the Lock Switch is unlocked when booting the PowerHub, the TTY ports use the default baud rates (9600 for TTY1 and 1200 for TTY2), regardless of the baud rates stored in NVRAM.

7.2.2 Removing and Replacing Interface Modules

Whenever it is necessary to remove a NIM in the PowerHub, the **`card-swap disable`** command must be issued so that the configuration manager can deactivate traffic to ports/segments on that NIM. Conversely, when replacing a NIM, of the same type in the same slot, the **`card-swap enable`** command reactivates traffic to the ports/segments on the NIM. These operations can be accomplished while the PowerHub is operating. Entering the **`card-swap`** command with no parameters displays a list of the NIMs currently installed in each card slot. Refer to the *PowerHub Hardware Reference Manual* for detailed procedures on removing and replacing NIMs.

NOTE

NIMs can only be swapped when the chassis contains at least one redundant power module. See the *PowerHub Hardware Reference Manual* for information about power redundancy. The **`card-swap`** command is only to be used with interface modules (NIMs).

NOTE

The NIM being installed must be of the same type as was removed. If a NIM is being replaced with a different type, power down the PowerHub, remove the card, insert the new card and then power on the PowerHub. This sequence loads the ID PROM information of the cards currently installed into the configuration manager.

The syntax for this command is:

```
card-swap|cs enable|disable <slot> card-swap|cs [show]
```

enable|disable Entering disable states that the card is being removed. Enable states that the card is being restored to the PowerHub.

<slot> Specifies which card slot is being enabled/disabled.

7.2.3 Displaying the Configuration

Issue the **config** command to display the current configuration. This command displays information on the physical configuration of the PowerHub. The following is an example of the **config** command. In this example, information is displayed for a PowerHub 7000 5-slot. Note that because no segments were specifically allocated for the PowerCell NIM, which starts with segment 02/07, only six virtual interfaces were configured. The PowerCell 700 supports up to 32 virtual interfaces.

```
3:PowerHub:system# config
Accelerator board is present. Accelerator IOP is being used.
Installed DRAM Size: 24 MB
tty1:  9600 baud
tty2:  19200 baud
PE:    slot 5
PM1:   present and good
PM2:   not present
PM3:   not present
PM4:   not present
  03/33 MM/MM
  02/17 UTP     UTP     UTP     UTP     UTP     UTP
        UTP     UTP     UTP     UTP     UTP     UTP
        UTP     UTP     UTP     UTP
  01/01 OC3-MF  OC3-MF  OC3-MF  OC3-MF  OC3-MF  OC3-MF
        OC3-MF  OC3-MF  OC3-MF  OC3-MF  OC3-MF  OC3-MF
        OC3-MF  OC3-MF  OC3-MF  OC3-MF  ----    ----
        ----    ----    ----    ----    ----    ----
        ----    ----    ----    ----    ----    ----
        ----    ----
4:PowerHub:system#
```

The **config** command displays the following information.

- Whether a Packet Accelerator is present on the Packet Engine and if the accelerator input/output processor (IOP) is in use or not.
- The amount of dynamic random access memory (DRAM) on the Packet Engine.
- The current baud rates for RS232 tty1 and tty2 ports.
- The slot occupied by the Packet Engine, indicated by PE:. In this example, the Packet Engine is in slot 5, the top slot in a 5-slot chassis.
- The presence and status of power module, indicated by PM1:, PM2:, PM3:, and PM4:.
- The slot number and starting segment number for each slot, and the media type in use in each segment position. The row beginning 01/01: displays the configuration of the NIM in slot 1, beginning with segment 1. The row beginning 02/17: shows the configuration in slot 2, beginning with segment 17, and so on. Remember that segments are numbered from left to right, bottom to top. The number of segments

in each NIM slot depends on how many are allocated in NVRAM. If slots are not explicitly allocated in NVRAM, the PowerHub defaults to the maximum allowable segments per NIM. Empty NIM slots are not displayed.

7.2.4 Setting and Displaying the System Time and Date

The **date** command is used to display or set the system date and time. The syntax for this command is:

```
date set [YYMMDD]hhmm[.ss] date [show]
```

set Sets the specified date and / or time.

[YYMMDD]hhmm> Specifies the year (*YY*), month (*MM*), day (*DD*), hour (*hh*), and minute (*mm*). To set the time, but not the date, specify <*hhmm*>**[.**<*ss*>**]**. (The software reads this argument from right to left, so any additional arguments can be specified with <*hhmm*>. For example, specifying <*DDhhmm*>, also specifies the day. Note that the arguments must be specified in the order shown. For example, <*YYhhmm*> or <*DDMMYYhhmm*> cannot be entered.)

[.ss] Optionally specifies the seconds. If this argument is used, make sure to use the period (**.**) in front of the seconds. If the number of seconds is not specified, the value is set to 00.

If either argument is not used, the current system date and time are displayed.

Here are some examples of the use of this command. In the first example, the current system time and date are displayed. In the second example, the time and date are changed. The new time and date are then displayed.

```
29:PHswitch:system# date
Wed Aug 27 11:27:15 1997
30:PHswitch:system# date set 9708271130.10
date set to: Wed Aug 27 11:30:10 1997
31:PHswitch:system# date
Wed Aug 27 11:30:15 1997
32:PHswitch:system#
```

7.2.5 Setting the Data Carrier Detect Parameter

The Data-Carrier Detect parameter can be displayed or changed using the **dcd-detection** command. Here are some examples of the command. The syntax for this command is:

dcd-detection|dcd enable|disable dcd-detection|dcd [show]

enable|disable Specifies whether to enable or disable data-carrier detection

Note that in this example, the short form of the **dcd-detection** command is used.

```
48:PHswitch:system# dcd
dcd-detection is currently disabled.
49:PHswitch:system# dcd enable
dcd-detection enabled
50:PHswitch:system#
```

7.2.6 Displaying the MAC Address of the PowerHub Switch

To display the Mac address of the PowerHub, issue the **ethaddr** command. The syntax for this command is:

ethaddr|ea [show]

Here is an example of the display produced by this command. In this example, the short form of the command is used.

```
9:PowerHub:system# ea
Ethernet address: 00-00-ef-03-9a-b0
10:PowerHub:system#
```

7.2.7 Displaying ID and Power Information

The Packet Engine and all types of NIMs contain a special PROM called the ID PROM. The ID PROM contains identification information and power requirements for the module. This information can be displayed using the **idprom** command. Here is the syntax for this command:

idprom|idp [show] *<slot number>***|all**

<slot number> Specifies the NIM slot containing the module.

Following is an example of the display produced by this command. In this example, information is displayed about the NIM in slot 2.

```
54:PHswitch:system# idprom 2

         Card Type: UTP 16x1 Interface Module
          Serial #: 632027371
             Model: 7202-00
          Revision: G
             Issue: 1
         Deviation: <not set>

  Power Requirements:
    5000 mA at 5V
55:PHswitch:system#
```

The ID PROM display shows the following information:

Card Type: Specifies the card currently installed in the slot specified.

Serial #: Displays the serial number of the specified card.

Model: Displays the model number of the specified card.

Revision: Displays the revision level of the card.

Issue: Displays the card issue number.

Deviation: If applicable, displays the factory-assigned deviation number. Only some modules have deviation numbers.

Power Requirements: Displays the maximum amperage (milliamps) required by the module at +12-volts, +5-volts, or +3.3-volts, as applicable.

Some older revisions of the Packet Engine and NIMs do not contain ID PROMs. If the **idprom** command is issued against such a module, or an empty card slot, the following message is displayed:

```
55:PHswitch:system# idprom 4
unable to read IDPROM information from slot 4
56:PHswitch:system#
```

7.2.8 Changing the Password

The **passwd** command is used to change the system password associated with “root” or “monitor” logins. The syntax for this command is:

```
passwd [root|monitor]
```

root|monitor Indicates the management capability for which the password is being changed.

To change a password:

1. Issue the **passwd** command, specifying the appropriate management level (root or monitor) capability. A prompt is displayed to enter the new password.
2. Enter the new password to be assigned to this management level. If no password is to be set, press Enter.
3. A prompt is then presented to Re-enter the password (`Re-enter new password:`) previously entered.
4. Re-enter the password that was entered at the `New password:` prompt, press Enter.

NOTE

This prompt is not displayed if the Lock Switch is in the unlocked position (U) or the Lock Switch jumper is set to Unlock. Instead, the New password: prompt is displayed.

5. The message “`Password changed`” is displayed to confirm that the password was changed.

The following example shows how a password for root management capability is changed.

```
52:PHswitch:system# passwd root
New password:*****
Re-enter new password:*****
Password changed
53:PHswitch:system#
```

For security reasons, the input shown above with asterisks does not appear when entered in response to the prompts. Remember that passwords are not required if the Lock Switch is in the unlocked (U) position. If the password is forgotten, turn the Lock Switch off, log in and enter a new password, then turn the Lock Switch on again.

7.2.9 Reading a Configuration File

When loading the software, the system looks for a configuration file on the default-device:

- If the Compact Memory module was used to load the software, the system looks for a file named cfg. If present, this file is automatically loaded and its configuration information is used to configure the PowerHub.
- If loaded from a BOOTP/TFTP server, the system looks for the configuration file specified in the bootdef (boot definition) file on the server.

Even if the system finds and loads a configuration file when the software is booted, additional configuration files can be loaded during a PowerHub session using the **`readcfg`** command.

NOTE

The new configuration information does not undo the configuration information contained in the cfg file. Instead, the new configuration is added to the current configuration, until the PowerHub is powered down or rebooted. The additional, or different, configuration information can be saved with the current configuration information by issuing the **`savecfg`** command.

7.2.9.1 Loading a Configuration from Compact Memory

To load a configuration file located in Compact Memory (or a saved on a terminal connected to a TTY port), issue the following command:

```
readcfg|rdcfg [-v] <file or device name>
```

-v Directs that each line of the configuration file be displayed to the console during execution.

<file or device name> Specifies a file name is to be read from the device specified. If no device is specified the default-device is assumed. For auto-configuration on boot up, use the file name 'cfg'. The last line in any configuration file must be the string 'endcfg' or 'ecfg'.

7.2.9.2 Loading a Configuration From a TFTP Server

To load a configuration file located on the default TFTP server, issue the following command:

```
readcfg|rdcfg [-v] [-h <host>] <remote-file>
```

-v Directs that each line of the configuration file be displayed to the console during execution.

-h <host> Specifies the IP address of the TFTP server. Unless a default TFTP server was specified using the tftp set command, this argument must be included. For information on the tftp set command, see Chapter 4 in the *PowerHub Software Manual, V 2.6* (Rev C).

<remote-file> Specifies the configuration file name. Specify a name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called fore and this directory is specified as the TFTP home directory, do not specify fore as part of the file name.

NOTE

The tftp version of the **`readcfg`** command only works if the system was last boot from a tftp boot server. Refer to the *PowerHub Software Reference Manual* for instructions on setting up a TFTP boot server.

7.2.10 Rebooting the PowerHub

The PowerHub can be loaded from a command-line session by entering the **`reboot`** command. The **`reboot`** command performs a cold restart of the PowerHub. During a cold restart, the Packet Engine conducts a power-on self-test to check its various hardware components. Following successful completion of the power-on self-tests, the PowerHub software is loaded.

7.2.11 Saving the Configuration

Configuration changes can be saved to the configuration file (**`cfg`**) in Compact Memory (or on the TFTP server), in which case they are automatically applied each time the software is loaded. Alternatively, changes can be saved to a different filename or to a device attached to TTY1 or TTY2. If configuration changes are saved to a file other than cfg, the file must be loaded after the software is loaded to apply the configuration settings to the PowerHub.

7.2.11.1 Saving Configuration Files to Compact Memory

To save a configuration file, issue the following command:

savecfg|svcfg *<file or device name>*

<file or device-name> A filename must be specified. If a device is not specified the configuration file is saved to the default-device.

In the following example, the current configuration is saved to a file named Lab1.cfg on the default-device.

```
98:PHswitch:system# savecfg Labl.cfg
99:PHswitch:system#
```

7.2.11.2 Saving the Configuration to a TFTP Server

To save a configuration file to a TFTP server, issue the following command:

savecfg|svcfg [-h *<host>***]** *<remote-file>*

-h <host> Specifies the IP address of the TFTP server, if different than the address specified with the **tftp set server** command. Unless a TFTP server was specified using the **tftp set server** command, include this argument. For information on the **tftp** commands, refer to Chapter 16 of the *PowerHub Software Reference Manual*.

<remote-file> Specifies the name of the configuration file to be saved. Specify a name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called fore and this directory is specified as the TFTP home directory, do not specify fore as part of the file name.

NOTE

The tftp version of the **readcfg** command only works if the system was last boot from a tftp boot server. Refer to the *PowerHub Software Reference Manual* for instructions on setting up a TFTP boot server.

NOTE

Some TFTP servers require that the remote file name exist on the server before writing to that file name. If the server requires that the filename already exist, create a short file (named the same as the configuration file) on the server, then specify that file name for <*remote-file*>.

NOTE

On some TFTP servers, including servers running Sun/OS 4.x, files overwritten on the server are not properly truncated. When overwriting an existing file on the TFTP server, if the older version of the file is longer than the new file, the older version is not truncated properly by the server. As a result, the new version of the file contains part of the older version of the file.

An example of the use of this command is.

```
99:PowerHub:system# tftp svcfg 147.128.128.7 Lab1.cfg
Configuration saved to Lab1.cfg
```

7.2.12 Setting and Displaying the System Location

The system location can be changed using the **syslocn** command. The syntax for this command is:

syslocn set *<location>* **syslocn [show]** *<location>*

set Sets the location specified.

<location> Specifies the location of the PowerHub. Any alphanumeric string up to 24 characters in length can be specified. The location name cannot contain blanks.

If a location is not specified, the current location name is displayed.

The following example shows how to display the current system location and to change the location variable. The new system location is defined as “Pittsburgh.” (Note that the system name is now “PHswitch,” as changed by the **sysname** command in the example above.)

```
26:PHswitch:system# syslocn
Current system location is: Undefined

27:PHswitch:system# syslocn set Pittsburgh
System location set to:
Pittsburgh
28:PHswitch:system#
```

7.2.13 Setting and Displaying the System Name

The system name is shown in the command prompt. The default system name is PowerHub. The system name can be changed using the **sysname** command. The syntax for this command is:

sysname set *<location>* **sysname [show]** *<location>*

set Sets the name specified.

<location> Specifies the name assigned to this PowerHub. Any alphanumeric string up to 24 characters in length can be specified. The name cannot contain blanks.

If a name is not specified, the current name is displayed.

The following example shows how to display the current system name and to change the name variable. The new system name is defined as “PHswitch.”

```
22:PowerHub:system# sysname
Current system name is: PowerHub
23:PowerHub:system# sysname set PHswitch
System name set to 'PHswitch'.
24:PHswitch:system#
```

7.2.14 Displaying the Temperature

The Packet Engine and all types of NIMs contain a temperature sensor that reads the temperature of the module with an accuracy of plus or minus 0.5° C. The current temperature of a module can be displayed by issuing the following command:

temperature|temp [show] *<slot number>***|all**

<slot number> Specifies the slot that contains the module for which to display the temperature.

all Displays the temperature for all installed modules.

In the example that follows, the first command displays the temperature for all modules. The second, displays the temperature for the module in slot 2.

```
11:PowerHub:system# temp all
slot  5, temp  41.5 degrees C
slot  3, temp  35 degrees C
slot  2, temp  32 degrees C
slot  1, temp   0 degrees C
12:PowerHub:system# temp 2
slot  2, temp  32 degrees C
13:PowerHub:system#
```

Note that the PowerHub is designed to operate over a range of external ambient temperatures. An additional temperature rise inside the chassis is accounted for in the design of the product.

Some older revisions of the Packet Engine and NIMs do not contain an ID PROM. If the **`temperature`** command is issued against a module that does not contain an ID PROM, or against a slot that does not contain a NIM, the system displays the following message:

```
13:PowerHub:system# temp 4
slot  4, temp not available
14:PowerHub:system#
```

7.2.15 Enabling/Disabling TTY2

Before setting or changing the baud rate associated with tty2 using the **`baud`** command (see Section 7.2.1), the tty2 port must be enabled using the **`tty2`** command. The syntax for this command is:

`tty2 enable|disable`

enable|disable Enables or disables the tty2 port.

An example of the use of this command is:

```
43:PHswitch:system# baud
TTY        Baud Rate
1         9600
2         19200
44:PHswitch:system# tty2 enable
45:PHswitch:system# baud set tty2 9600
Changed tty2 baud rate to 9600; written to nvram
46:PHswitch:system# tty2 disable
tty2 is now closed
47:PHswitch:system#
```

NOTE

If the Lock Switch is unlocked when booting the PowerHub, the TTY ports use the default baud rates (9600 for TTY1 and 1200 for TTY2), regardless of the baud rates stored in NVRAM.

NOTE

Before the baud rate can be set for TTY2, a session must be opened on the port. To open a TTY2 session, issue the **`tty2 enable`** command.

7.2.16 Displaying the System Uptime

The **`system uptime`** command displays how long much time has elapsed since the last time the PowerHub software was loaded. There are no parameters on for the **`uptime`** command. Here is an example of the **`uptime`** command.

```
15:PowerHub:system# uptime
Elapsed time since last reboot: 3 hours, 9 minutes, 9 seconds
16:PowerHub:system#
```

7.2.17 Displaying the Software Version

The version command displays the version level of software currently running on the NIMs and Packet Engine installed in the PowerHub. The syntax of this command is:

```
version|ver [show] [<slot-number>|all]
```

<slot-number> Lists the version of software on the card in the slot specified.

all Lists the version of software on the Packet Engine and all Intelligent NIMs installed in the PowerHub switch. If the all parameter is not used, software version information is displayed for the Packet Engine only.

A typical display of the use of this command is shown below.

```
61:PHswitch:system# ver

         Card Type: Packet Engine - 40MHz
          Serial #: 633020265
             Model: 7101-01
          Revision: C
             Issue: 2
         Deviation: <not set>

  PowerHub Version: PH 7pe FT4.0.0-Alpha51 (@5567) (s#5) 1997.08.23 17:14
      PROM Version: 7pep-2.5.5 (s1.85) 1996.06.21 11:14
62:PHswitch:system#
```

Card Type: Specifies the card currently installed in the slot specified.

Serial #: Displays the serial number of the specified card.

Model: Displays the model number of the specified card.

Revision: Displays the revision level of the card.

Issue: Displays the card issue number.

Deviation: If applicable, displays the factory-assigned deviation number. Only some modules have deviation numbers.

PowerHub Version: Displays the PowerHub type, software version installed, with the software build number and date and time of build.

PROM Version: Displays the PROM version information which includes the version number with build and date and time of build.

7.2.18 Displaying Boot Information

The PowerHub software can be loaded from floppy diskette, Compact Memory, or a file server (netbooting). After the software is loaded, the following information is logged in memory as the bootlog. This information can be retrieved by executing the **`bootinfo`** command. The bootlog contains:

- The date and time the system was started.
- The date, time and nvram bootorder (see *Chapter 9, NVRAM Subsystem* on setting the nvram boot order).
- The boot device used to boot. The value can be f (floppy diskette or Compact Flash Card), `m` (Flash Memory Module) or n (network). This value shows the boot source actually used, which may differ from the boot order specified in NVRAM.

The **`bootinfo`** command displays the contents of the bootlog. An example of the **`bootinfo`** command is.

```
7:PowerHub:system# bootinfo
Wed Aug 27 08:00:57 1997 start
Wed Aug 27 08:01:02 1997 nvram boot order: m
boot device: m
8:PowerHub:system#
```

7.2.19 System Configuration Commands

The system subsystem contains commands that allow the changing of configuration parameters. The following sections provide more information on changing configuration information.

7.2.19.1 Rebooting Without Loading the Default Configuration File

When rebooting the software, the system looks in the bootdef file for the name of a configuration file. If the software needs to be loaded without loading the configuration file, use the global command **`rename`** command to temporarily rename the default configuration file before loading. For example, if the boot definition file calls the configuration file cfg, rename cfg to temp.cfg. The syntax for this command is:

`mv|rename` *`<file1> <file2>`*

<file 1> Specifies the current filename.

<file 2> Specifies the new filename.

After loading the software, restore the configuration file to its original name using the same command. When loading a second time, the switch processes the `cfg` file and boots with the default configuration.

7.2.19.2 Editing a Configuration File

Although changes to a configuration file can be automatically written using the **savecfg** command, the configuration file can be edited manually using a text or ASCII editor. To move the file to a machine that contains an ASCII editor:

- If the editor is on a file server, use the **tftp put** command to write the file to the server, edit the file, then use the **tftp get** command to download the edited file back into the Flash/Compact Memory Module. (See Chapter 4 in the *PowerHub Software Manual, V 2.6* (Rev C) for information about **tftp** commands.)
- If the editor is on a PC or Macintosh, use the ZMODEM **zw** command to write the file to the PC or Macintosh, edit the file, then use the **zr** command to transfer the edited file to the Flash/Compact Flash module. These commands are available from the boot PROM prompt, <PROM-8pe>.

7.2.19.3 Capturing Configuration Information

The system software diskettes contain a file that enables configuration information about the PowerHub system to be captured. The file is called dispcfg. When reading the file (using the **`readcfg dispcfg`** command), the commands in the file display configuration information on the management terminal.

If problems are experienced, FORE Systems TAC (Technical Assistance Center) might request that the dispcfg file be read so that the information captured by the file may be helpful in resolving the problems. To read the dispcfg file, issue the following command:

```
system readcfg dispcfg
```

CHAPTER 8 Media Subsystem

This chapter describes the media subsystem commands. The media subsystem commands relate to the physical media and bridging configuration information. This chapter explains the commands that allow the following:

- Show bridging-related configuration information
- Clear, enable, disable or show Inter-Segment statistics
- Set or show Ethernet NIM LED modes
- Set, clear or show Port Monitoring
- Set or show Ethernet controller operating mode
- Enable, disable or show port-by-port statistics
- Enable or disable transmission and reception of packets on given segments
- Set or show segment names for given segments
- Enable, disable or show automatic segment state detection for all or specified segments
- Set segment state detection thresholds for specified segments
- Show port-level status for UTP ports
- Show or clear media-level statistics for UTP ports or segments

8.1 Displaying Bridge-Related Configuration

The current port and segment configuration can be displayed using the **config** command. The **config** command is issued from the media subsystem and displays the following bridge-related information:

- Port monitoring
- Forwarding status of segments
- UTP port receiver enable / disable status
- Automatic segment state detection
- Segment names
- Port level statistics collection
- Inter-segment statistics collection

The syntax of the **config** command is:

config [show] [*<params>*] [*<disp-restrictors>*]

<params> Specifies a comma separated list of

monitor	show lan-monitor status
segment	forwarding enabled or not
[port]receive	receivers enabled or not
ssd	segment-state detection
[segment]names	names assigned to segments
portstats	collection enabled or not
isstats	collection method enabled or not

<disp-restrictors> Specifies the segment or segments to display status on in a segment list <seglist>.

Entering **`config 2.1, to display the bridge-related configuration information of port 2.1,`** displays the following:

```
24:PHswitch:media# config 2.1
Port Monitoring
---------------
Packets...
not being monitored on segment 2.1
Forwarding status of segments
-----------------------------
2.1 :enabled
UTP port receiver enable/disable status
---------------------------------------
Slot  3:  .
Slot  2:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Slot  1:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .
Automatic segment state detection
---------------------------------
Segment  2.1 : enabled (currently good)
Segment names
-------------
2.1 : Port_17
Port level statistics collection: currently disabled.
Inter-Segment Statistics collection is disabled
25:PHswitch:media#
```

8.2 Inter-Segment Statistics

Inter-Segment Statistics can be cleared, enabled, disabled or displayed using the **isstats** command. The **isstats show** command displays the packet statistics of packets between segments of the installed PowerHub NIMs, if statistic collection is enabled. The syntax of **isstats** is:

isstats [show] [*<params>*] [*<disp-restrictors>*] isstats clear|enable|disable

<params> Specifies a comma separated list of p,o for packets and/or octets.

<disp-restrictors> Specifies a fr[om]=<seglist> to=<seglist> list of segments.

The following display shows two variations of displaying inter-segment statistics for either a range of segments or specific segments. Note that a range is separated with a hyphen.

```
12:PHswitch:media# isstats show 1.1-1.4
Segment to segment statistics collection is disabled
FROM        TO>          1.1          1.2          1.3          1.4
1.1  :(pkts)             (0)          (0)          (0)          (0)
     :octets              0            0            0            0
1.2  :(pkts)             (0)          (0)          (0)          (0)
     :octets              0            0            0            0
1.3  :(pkts)             (0)          (0)          (0)          (0)
     :octets              0            0            0            0
1.4  :(pkts)             (0)          (0)          (0)          (0)
     :octets              0            0            0            0
14:PHswitch:media# isstats show 1.1,2.1
Segment to segment statistics collection is disabled
FROM        TO>          1.1          2.1
1.1  :(pkts)             (0)          (0)
     :octets              0            0
2.1  :(pkts)             (0)          (0)
     :octets              0            0
15:PHswitch:media#
```

8.3 Ethernet LED Modes

The operating mode information displayed by the LEDs on installed Ethernet modules can be set to reflect either transmission collision and activity or transmit and receive activity. The `ledmode` command configures the operating mode of the traffic LEDs (C/X and A/R) on all types of Ethernet modules except the UEM (Universal Ethernet Module). LEDs are set as a group for the entire module, not individually on a segment-by-segment basis.

Set the LEDs to C and A (transmission collision and activity) or to X and R (packet transmit/ receive). The LNK LED cannot be configured and always shows link status information for the corresponding segment. (For a description of the information indicated by each setting, see the *PowerHub Hardware Reference Manual*.)

NOTE The traffic LEDs on the Universal Ethernet Module (UEM) cannot be configured. The LEDs on the UEM always indicate receive, transmit, and collision separately.

The syntax for the **ledmode** command is:

```
[show]|set ledmode|lm [<slot>] ca|xr
```

show|set Indicates whether to show current configuration or change the current configuration.

slot Indicates the NIM slot that contains the Ethernet module with the LEDs being configured.

If a NIM slot is specified, but not an LED setting, the current setting for the specified module is shown. If a NIM slot is not specified, the current LED settings for all Ethernet modules (except UEMs) in the chassis are shown.

ca|xr Configures the LEDs:

ca Reflects transmit collision and activity (transmit and receive).

xr Reflects packet transmit and receive activity. The default is **xr**.

The default **xr** configuration should be used except in networks that experience a large number of collisions. Note that collisions do not occur on segments that are configured for full-duplex operation. Entering **`ledmode`** with no additional parameters displays the following.

```
2:PHswitch:media# ledmode
Slot 01: LED not configurable on this module type.
Slot 02: xr (leds reflect transmit and receive activity)
Slot 03: LED not configurable on this module type.
3:PHswitch:media#
```

Setting **`ledmode`** to ca displays the following.

```
4:PHswitch:media# ledmode set 2 ca
Slot 02: ca (leds reflect collision and activity)
```

8.4 Port Monitoring

Port monitoring allows the use of a protocol analyzer (such as a Sniffer, LANalyzer, or Network Pharaoh) connected to a PowerHub segment and to monitor the traffic on any other segment or set of segments. Rather than separately attaching the analyzer to each segment to be monitored, the analyzer can be attached to one segment, then Port Monitoring commands in the PowerHub software can be used to select the segments to monitor.

The segments being monitored can be changed without moving the analyzer to another segment. In addition, traffic on more than one segment can be monitored simultaneously, without the need to use multiple analyzers on multiple segments.

The **`monitor`** command is used to enable port monitoring on the PowerHub. To enable port monitoring perform the following:

1. Log on with root capability.
2. Disable forwarding on the segment to which the analyzer is attached by issuing **`segment pdisable`** <*`seg-list`*>, where <*`seg-list`*> specifies the segment the analyzer is to be attached (refer to Section 8.8 for more information on the **`segment`** command).

NOTE

If it is found that the Port Monitoring feature is being used frequently, configure one segment to Port Monitoring to prevent having to disable forwarding each time this feature is used.

3. Enable port monitoring with the **`monitor set`** command (refer to Section 8.4.4 for descriptions of the options and parameters available for use with the **`monitor set`** command).

8.4.1 How Port Monitoring Works

Conceptually, port monitoring "copies" packets from the monitored segments to the monitoring segments. Actually, packets are not really copied because this would dramatically reduce performance. Instead, a pointer to the packet buffer containing a monitored packet is placed on the transmit queue for the monitoring segment(s), and the packet buffer is freed up only after it has been transmitted both to its normal destination, if any, and to the monitoring segment(s). For monitoring purposes, packets are classified into three types:

Incoming A packet that is received on the monitored segment. An incoming packet might or might not be forwarded, according to the usual bridging and routing rules.

Forwarded A packet received on one segment, then transmits on the monitored segment.

Generated A packet transmitted on the monitored segment as required by the internal protocol stacks. This includes outgoing TCP packets in TELNET sessions, UDP packets for RIP updates and SNMP replies, ARP requests and replies, ICMP packets for various IP routing errors, Spanning-Tree hello and topology-change packets, and various packets generated by the IPX, AppleTalk, and DECnet protocol stacks.

Port monitoring monitors packets regardless of any filters defined on the monitoring segment. This includes any filters that normally block traffic from the monitored segment to the monitoring segment. Filters defined on the monitored segment do remain in effect. In addition, incoming packets are monitored regardless of a segment's Spanning-Tree state (blocked or forwarding) or the enabled state (enabled or disabled) of the monitored segment.

8.4.2 Performance Considerations and Operation Notes

In general, port monitoring does not adversely affect PowerHub performance on the monitored segments or other segments. However, if the monitored traffic load is greater than the capacity of the monitoring segment, then not all monitored packets are successfully queued. Packets not queued onto the monitoring segment for this reason are still delivered to their normal destinations.

When multiple segments are monitored, packets from all segments are queued onto the monitoring segment in the approximate order in which they were received, forwarded, or generated. Note that if a packet is "incoming" on one monitored segment and "forwarded" on another monitored segment, only one copy of the packet is queued onto the monitoring segment.

When outgoing (forwarded or generated) and incoming packets are monitored on a segment, they might not appear on the monitoring segment in the same order in which they appear on the monitored segment. This can happen because an outgoing packet is queued for transmission on the monitoring segment at the same time that it is queued for transmission on the monitored segment, not when it is actually transmitted. Therefore, it is possible for one or more packets to be received on the monitored segment and queued up after the outgoing packet on the monitoring segment, even though they appear on the monitored segment before the outgoing packet is actually transmitted. However, the order of packets within either the incoming stream or the outgoing stream on the monitored segment is preserved on the monitoring segment.

NOTE

Incoming runt packets, giant packets, and packets with FCS or frame-alignment errors are not monitored. Long (larger than 1518 bytes) packets on FDDI segments are fragmented and the fragments appear on the protocol analyzer. This applies even if the monitoring segment and monitored segment are both FDDI segments.

NOTE

Do not use the monitoring segment for routing or any other purpose except monitoring. The monitoring segment should not have any devices connected to it other than a protocol analyzer. Other types of connected devices (workstations, servers, and so on) can get very confused by packets from monitored segments.

8.4.3 Packet Modifications

During normal bridging and routing, certain packets are modified before being forwarding. For example, both the MAC-layer and network-layer (routing) headers in routed packets are modified. Moreover, when packets are forwarded from FDDI to Ethernet, or vice versa, it modifies the packets accordingly.

With port monitoring, the modified packet, not the original packet, is transmitted to the monitoring segment. As a result, the packet displayed by the protocol analyzer is the modified packet. The way the packet is modified depends upon the segment type (Ethernet or FDDI) and the forwarding algorithm used, as summarized in Table 8.1.

Table 8.1 - Packet Modifications On Monitoring Segment

Traffic Type	Monitored Segment Type	Monitoring Segment Type	Packet Is...
Bridged Description: Forwarded, or incoming but not forwarded.	Ethernet	Ethernet	U
	Ethernet	FDDI	T
	FDDI	Ethernet	T
	FDDI	FDDI	TT/U
Routed Traffic Description: Forwarded.	Ethernet	Ethernet	M, I, R
	Ethernet	FDDI	M, I, R, T
	FDDI	Ethernet	M, I, R, T
	FDDI	FDDI	M, I, R, TT/U
Routed Traffic Description: Incoming but not forwarded.	Ethernet	Ethernet	I
	Ethernet	FDDI	I, T
	FDDI	Ethernet	I, T
	FDDI	FDDI	I, TT/U
Generated Traffic Description: Generated.	Ethernet	Ethernet	U
	Ethernet	FDDI	T
	FDDI	Ethernet	U
	FDDI	FDDI	U

KEY:I=IP TTL and checksum changed
M=MAC address changed
R=Routing header changed
U=Unmodified
T=Translated
TT/U=Double Translated but Unchanged.			

The modifications made to packets appearing on the monitoring segment are further explained by the following key:

U The packet is not changed in any way. If the packet also undergoes a double-translation (denoted in Table 8.1 by TT), this means the packet is double-translated, but the resulting packet is identical to the packet before double translation.

M The destination MAC address is changed to the address of the next hop. The source MAC address is changed to the address of the PowerHub.

I If the packet is an IP packet, certain fields in the IP header are changed. Specifically, the TTL field is decremented and the IP-header checksum is incremented. The IP header and payload are otherwise unmodified.

R Certain fields in the network header (for example, the IP header) might be changed, depending upon the routing protocol:

AppleTalk The hop count is increased by one. Also, if the packet contains a checksum, the checksum is changed appropriately.

IP If the packet has an options field specifying source routing or route tracing, the appropriate modifications are made. If IP security options (RFC 1108) are used, then option fields may be added to or removed from the header. In rare cases, adding option fields causes the packet to exceed 1518 bytes, and consequently become fragmented.

IPX The only IPX field that is changed in the header is the “Transport Control” field. This field is incremented by 1 for each router that the packet passes through. (This field is similar to the TTL field in the IP header). Note, however, that the MAC header can change in many different ways.

In the simplest case, where there is no header translation, the MAC header is changed as follows:

src-mac-addr Changed to address of the PowerHub.

dst-mac-addr Changed to the address of either the destination node or the next hop gateway.

When header translation is involved, in addition to the two fields above, the header type changes from the configured type for the receiving network to the configured type for the destination/next-hop network. The four different encapsulation types used for IPX are IEEE 802.3 (Raw), IEEE 802.2 (LLC), IEEE 802.2 (SNAP) and Ethernet-II.

DECnet The following fields in the MAC header are changed:

src-mac-addr Changed to address of this router.

dst-mac-addr Changed to the address of either the destination node or the next hop gateway.

In addition, a change is made to the DECnet long data packet headers (these are normal data packets). The long data header contains a “flags” field which is modified as follows:

If the source and destination nodes of the packet are both on the same segment, the “INTRA ETHERNET PKT” bit is set.

If the source and destination nodes are on different segments, the “INTRA ETHERNET PKT” bit is cleared.

If the destination node is not reachable and the sender has set the “RETURN TO SENDER REQ” bit, the PowerHub clears this bit and sets the “RETURNING TO SENDER” bit. In this case, the MAC header is changed as follows:

src-mac-addr Changed to the PowerHub MAC-layer hardware address.

dst-mac-addr Changed to the MAC-layer hardware address of the sender.

T The packet undergoes translation between Ethernet and FDDI formats. In the case of long FDDI IP packets (larger than 1518 bytes), the packet also undergoes IP fragmentation. (Long non-IP packets are not monitored.)

TT/U The packet undergoes a double translation, from FDDI to Ethernet and back. The end result normally appears unchanged, except for fragmentation in the case of long IP packets. (Long non-IP packets are not monitored.)

8.4.4 Monitoring a Segment

Use the **monitor** command to monitor one or more segments. Before monitoring one or more segments, disable forwarding on the segment to which the monitored traffic is being sent. Use the **segment** command to disable forwarding on a segment. See Section 8.8 for more information about the **segment** command. After disabling the segment, issue the **monitor** command to begin monitoring. If multiple **monitor** commands are issued, their effect is cumulative. That is, the PowerHub monitors **all** of the traffic specified by **all** of the commands. The syntax for the **monitor** command:

```
monitor set [from <monitor-spec>] [to <monitor-spec>] on <seglist>
          monitor [show] [<seglist>] monitor clear
```

set Sets monitoring on the specified segment or segment list (<*seglist*>).

From <monitor-spec> Specifies which segments are to be monitored. Packets entering the switch through segments identified by this variable are copied to the monitoring segment. The <monitor-spec> can be one of the following options:

internal	packet from internal protocol stacks.
<*seglist*>	packets from the segments listed
*	packets from all segments
any	packet from internal protocol stacks.This option is the default and is the same as the “internal” option.

to <monitor-spec> Specifies which segments are to be monitored. Packets leaving the switch through segments identified by this variable are copied to the monitoring segment. The <monitor-spec> can be one of the following options:

internal	packet to internal protocol stacks.
<*seglist*>	packets to the segments listed

* packets to all segments

any packet from internal protocol stacks.This option is the default and is the same as the “internal” option.

<seglist> Indicates the segments to which the monitored traffic is to be sent. For most applications of Port Monitoring, this segment list contains just one segment.

[show] Displays the current port monitoring configuration.

clear Clears port monitoring parameters previously set.

NOTE Packets destined for internal protocol stacks cannot be differentiated from other incoming packets. To exclude packets destined for internal protocol stacks, use the “*” option in the “**[to** *<monitor-spec>***]**” field and filter by MAC address with external monitoring equipment.

8.5 Operating-Mode

The **`operating-mode|om`** command is used to set the operating mode of an installed Ethernet controller. The **`operating-mode`** command can only be used to set the operating mode on AUI, 10Base-T, 10Base-FL, 100Base-TX and 100Base-FX ports.

```
24:PHswitch:media# om show 2.1
Segment  2.1 : fdx
25:PHswitch:media# om show 2.1-2.5
Segment  2.1 : fdx
Segment  2.2 : fdx
Segment  2.3 : fdx
Segment  2.4 : fdx
Segment  2.5 : fdx
26:PHswitch:media# om show 2.1,2.3,2.5
Segment  2.1 : fdx
Segment  2.3 : fdx
Segment  2.5 : fdx
27:PHswitch:media#
```

The syntax for the **operating-mode|om** command is:

operating-mode|om [show] [*<seglist>***|all]**
operating-mode|om set *<seglist>***|all fdx|lbk|flbk|declbk|normal|hdx**

[show] Displays the current operating-mode settings of the specified ports or all.

set Set the operating mode for the ports specified to the mode specified.

<seglist>|all Specifies a segment, segment list, or all, that the operating mode is to be changed. The specified segments can be a range of segments separated by a hyphen or a comma-separated list of specific segments.

fdx|lbk|flbk|decblk|normal|hdx Set the specified segment(s) to the desired to the desired mode of operation.

Attempting to set a port to an operating mode that is not supported by that particular segment results in an error message as shown below.

```
27:PHswitch:media# om set 2.1 hdx
Segment  2.1 : media adapter only supports full duplex
28:PHswitch:media# om set 2.1 normal
Segment  2.1 : normal
29:PHswitch:media# om set 2.1 fdx
Segment  2.1 : fdx
30:PHswitch:media#
```

8.6 UTP Port Receiver Status

The **portreceive|pr** command enables or disables the receivers on specified UTP ports and can control each port independently. Enabling, or disabling, the UTP port receivers has no affect on the port transmitters. The syntax for the **portreceive|pr** command:

portreceive|pr penable|pdisable *<port-list>*

penable|pdisable Enables or disabled port receiver on the specified ports.

<port-list> Specifies a port, port list, or range of ports to be enabled or disabled. This variable can be a port name, a comma separated list of ports, or a dash-separated range of ports.

Following are some examples of this command:

```
296:PowerHub:media# portreceive pdisable 1.1-1.6
OK
297:PowerHub:media#
```

In the example above, the receive ports numbered 1 through 6 on segment 1 are disabled. In the example below, a list of segments are enabled. Note that no spaces are required after the commas.

```
296:PowerHub:media# portreceive penable 1.1,1.2,1.3,1.5
OK
297:PowerHub:media#
```

To show the current status of the UTP ports, issue the **show config portreceive** command. Here is an example of this command.

```
41:PHswitch:media# show config portreceive
UTP port receiver enable/disable status
---------------------------------------
Slot  3:  .
Slot  2:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Slot  1:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .
42:PHswitch:media#
```

8.7 Displaying Port-Level Statistics

Port-by-port statistics of 4x4, 4x6 or 2x8 Ethernet modules can be displayed using the **portstats** command. Before displaying port-level statistics, the **portstats enable** command must be issued. The syntax of the **portstats** command is:

portstats [show] [<*display-restrictor*>] portstats enable|disable

show Displays the port statistics for the segment(s) specified by <display-restrictor>. If no display-restrictor is indicated, all the portstats are listed.

<display-restrictor> Specifies the segment, list of segments, or range of segments for which statistics are to be displayed.

enable|disable Specifies whether to enable or disable statistics collection on the switch. When portstats is disabled, the statistics are reset to zero.

NOTE Enabling the collection of port-by-port statistics places an extra load on the PowerHub processors. It is recommended that port-by-port statistics be left disabled, except when necessary.

8.8 Configuring Packet Forwarding on Segments

Under certain circumstances it is desirable to disable the transmission of packets and the reception of packets on a specific segment. For example, when using the port monitoring feature of the PowerHub, it is not desirable for the segment receiving the monitored packet to receive or transmit any other traffic. To enable or disable packet forwarding and reception, use the **`segment`** command. The syntax for the **`segment`** command is:

`segment penable|pdisable` *`<segment-list>`*

<segment-list> Specifies a single segment, dash-separated segment range, or a comma-separated list of segments.

Here are some examples of this command:

```
195:PowerHub:media# segment pdisable 2.21-2.28
Segment 2.21: disabled
Segment 2.22: disabled
Segment 2.23: disabled
Segment 2.24: disabled
Segment 2.25: disabled
Segment 2.26: disabled
Segment 2.27: disabled
Segment 2.28: disabled
196:PowerHub:media#
```

In the above example, a range of segments, 2.21 to 2.28, is disabled. In the example below, a comma-separated list of segments is enabled. Note that there are no spaces in the comma-separated list.

```
198:PowerHub:media# segment penable 1.1,1.3,1.5,1.6
Segment 1.1: enabled
Segment 1.3: enabled
Segment 1.5: enabled
Segment 1.6: enabled
199:PowerHub:media#
```

8.9 Configuring Segment Names

When the PowerHub first boots and assigns segments, the segment numbers are assigned from bottom to top and are named starting with “Port_1.” To rename or display the port names on the PowerHub, issue the **segmentname** command.

Here is the syntax for the **segmentname** command:

segmentname|name sset *<name> <seglist>* **segmentname|name [show] [***<seglist>***]**

sset Sets the segment name for the specified segment(s).

show Displays the segment name(s) for the specified segment(s).

<name> Specifies the name to use as a replacement for the default “Port_*x*” name. This variable is not required when using the **show** argument.

<seglist> Specifies the segment number of the segment to be renamed. This variable must be a single segment number when renaming a segment. This variable may be a dash-separated range or a comma-separated list when used with the show argument. When the show argument is used and the <seglist> variable is not used, all segments are displayed.

An example of the **segmentname** command is shown below:

```
49:PHswitch:media# segmentname sset tpubs 2.1
Segment 2.1 named: tpubs
51:PHswitch:media# segmentname show 2.1
Segment names:
2.1 : tpubs
52:PHswitch:media#
```

8.10 Automatic Segment-State Detection

Automatic segment-state detection recognizes is a segment is down and automatically disables bridging and routing on that segment. When it has been detected that a segment’s state has changed, the segment is disabled (taken out of service) and the software is marked to denote the change. The updated segment state is displayed when the **ssd** command is issued.

NOTE

If automatic segment-state detection on a segment is disabled, the segment's state is always reported as "good" and interface states are always reported as "up" in the software. For information about a segment's or interface's state, enable automatic segment-state detection for that segment.

The method used to determine whether a segment is down differs depending upon the type of segment. Table 8.2 lists the methods used to determine the state of each type of segment.

Table 8.2 - Segment-State Detection Methods

Segment Type	**Segment is Determined To Be Down If...**
10Base-FB	No link-test pulses are present on this segment.
10Base-FL	No link-test pulses are present on this segment.
100Base-FX	No data or idle symbols are being received on this segment.
100Base-TX	No data or idle symbols are being received on this segment.
ATM	The ELAN goes down or the physical link to the AMA goes down.
AUI	No packets are received and a "loss of carrier" is detected T times over a 1-second period, where T is specified by the <*threshold*> argument on the **ssdthreshhold** command. Note that the PowerHub does not send test packets, but relies on client and network management traffic to detect carrier loss. The state is changed to "up" if at least one packet is received. When automatic segment-state detection is first enabled (for example, when the PowerHub is booted), each AUI segment begins in the "down" state but is changed to the "up" state as soon as it receives packets.
BNC/BNCT	No packets are received and transmit-buffer errors occur over a T-second time period, where T is specified by the <*threshold*> argument on the **ssdthreshhold** command. The state is changed to UP if at least one packet is received. As with AUI segments, when automatic segment-state detection is first enabled (for example, when the PowerHub is booted), each BNC segment begins in the "down" state but is changed to the "up" state as soon as it receives packets.
FDDI	The attachment configuration of the segment is "isolated."
MAU	No AUI cable carrying +12-volt current (standard for AUI) is connected to the MAU.
UTP	No link-test pulses are present on this segment.

8.10.1 Software Behavior When Disabled

When a segment is disabled, no packets are bridged or routed on that segment. Bridging and routing do not occur whether the segment is disabled by automatic segment-state detection or by issuing the **`segment pdisable`** command. See Section 8.8 for information on the **`segment pdisable`** command.

8.10.2 Default Setting

The default setting for the automatic segment-state detection differs depending upon the segment type. Table 8.3 lists the default setting for each segment type.

Table 8.3 - Automatic Segment-State Detection Default Settings

Segment Type	Default
10Base-FB	Enabled
10Base-FL	Enabled
10Base-T (UTP)	Enabled
100Base-FX	Enabled
100Base-TX	Enabled
ATM	Enabled
AUI	Disabled
BNC/BNCT	Disabled
FDDI	Enabled
MAU	Enabled

As shown in Table 8.3, all segment types except AUI and BNC/BNCT have automatic segment-state detection enabled by default. In general automatic segment-state detection should be left at the factory default settings.

8.10.3 Disabled on AUI, BNC, and BNCT

When automatic segment-state detection is enabled, the AUI, BNC, and BNCT segments are not enabled until the segments receive traffic. In most configurations, the AUI, BNC, and BNCT segments are connected to devices that are prepared to generate traffic. However, connecting AUI, BNC, and BNCT segments to AUI, BNC, and BNCT segments on another PowerHub, and the other PowerHub has automatic segment-state detection enabled on these segments, no traffic is exchanged by the segments. Each end of the segment waits to receive traffic before becoming enabled. As a result, neither end of the segment becomes enabled, and no traffic is exchanged.

If the device at the other end of the AUI, BNC, or BNCT segment is prepared to generate traffic, enable automatic segment-state detection on the segment. When the segment receives traffic from the other device, the segment is enabled.

8.10.4 Segment-State Detection on 10Base-T

Automatic segment-state detection should be enabled for all 10Base-T (UTP) segments, even if they are not going to be used. If automatic segment-state detection is disabled, the Ethernet controllers on the corresponding 10Base-T segments do not stop using the forwarding buffer for those segments. Instead, they fill their transmit buffers, even though no traffic needs to be forwarded. Full buffers can negatively affect packet throughput.

8.10.5 Explicitly Disabling Unused Segments

For AUI, BNC, and BNCT segments, heuristics are used to determine the segment state. Occasionally, electronic noise can make an AUI, BNC, or BNCT segment appear active when it is not. When this occurs, automatic segment-state detection believes the segment is active and does not disable it. Accordingly, it is recommended that segments be explicitly disabled when removing them from service. See Section 8.8 for information about the **segment** command.

8.11 Setting Segment-State Threshold

Segment-state thresholds can be set for AUI and BNC segments. To set segment-state detection thresholds for AUI and BNC segments, issue the **ssdt** command. The syntax for this command is:

```
ssdthreshold|ssdt sset <value> <seglist>
```

<value> For AUI segments, specifies the "loss-of-carrier threshold"; that is, the number of times a loss of carrier must be detected in a one-second period for the segment to be considered down and therefore by disabled by software.

For BNC segments, specifies the “idle period threshold”; that is, the number of seconds during which the segment must remain idle to be considered down and therefore be disabled by software.

<seg-list> Specifies the segment(s) for which automatic segment-state detection is to be enabled or disabled. If all is specified, the detection state is changed for all segments in the chassis.

NOTE

Automatic segment-state detection must be enabled on all 10Base-T segments. The Ethernet controllers refuse to transmit packets on any segment that does not have a “good” link status. As a result, buffers can become “stuck” on the output queue of 10Base-T segments that do not have a “good” link status. This can adversely affect the performance of the rest of the hub. By enabling automatic segment-state detection, buffers can be prevented from being enqueued on the segments, and allow any enqueued buffers to be recovered if the segments go down.

An example of setting the segment-state detection threshold for port 2.1 is shown below:

```
82:PHswitch:media# ssdt sset 10 2.1
Segment  2.1  : enabled (currently bad)
83:PHswitch:media#
```

This display shows information appropriate to each segment type. Because BNC segments are determined to be down if they are idle for the period specified by <*threshold*> (in this case, 5 seconds), their “idle period threshold” is shown. Because AUI segments are determined to be down if a “loss of carrier” is detected the number of times specified by <*threshold*> in a one-second period, the “loss of carrier” threshold (in this case, 10 seconds) is listed.

The other types of Ethernet segments are determined to be down in the absence of regular link-test pulses, or data or idle symbols. FDDI segments are down if the attachment configuration of the segment is “isolated.” ATM segments are down if the ELAN on the segment goes down or the physical link to the PHY goes down. In these cases, no threshold is shown.

8.12 Status

The **status** command is used to display the port-level status of the specified, or all, UTP ports on NIMs installed in the PowerHub. As shown in the below example, the Link Test, Partitioning and Polarity of all UTP ports is displayed.

```
11:PHswitch:media# status
Link Test:
Slot  3:  -
Slot  2:  Y  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
Slot  1:  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -
-  -  -  -  -  -  -  -
Partitioning:
Slot  3:  .
Slot  2:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Slot  1:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .
Polarity:
Slot  3:  .
Slot  2:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Slot  1:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
.  .  .  .  .  .  .  .
12:PHswitch:media#
```

The syntax for the **status** command is:

status [show] [*<params>*] [*<display-restrictors>*]

w<params> Specifies a comma-separated list of link, partition, polarity or all.

<disp-restrictors> Specifies the segment or segments to display the port-level UTP port status.

8.13 Media-Level Statistics

The **stats** command is used to display media-level statistics. If port statistics are being collected (refer to Section 8.7) they are displayed in place of segment-wide statistics unless the -s flag is used. Segment-level statistics can be cleared but not disabled. When segment-level statistics are cleared, they are reset to zero and immediately begin to increment as packets are received and sent by the switch. The syntax of the **stats** command is:

stats [show] [-p|-s] [*<params>*] [*<display-restrictor>*]
stats clear

-p Display port statistics, if available.

-s Display segment-level statistics.

<params> Specifies a comma-separated list of the following (or “all”)

pi, po, oi, oo, bpi, bpo, pu, rbe, xbe, fcs, fa, c, rc, tc, q, gp, cu, lc, er, tm

In addition to above parameters, the following apply specifically to 2x8 Fast Ethernet Repeaters

ce, drm, jbrs, se, rts, lcs, ap, iss, sfc, lsa

See Table 8.4 for parameter definitions.

Table 8.4 - Segment Level Statistic Parameters

Parameter	Description
pi	packets in
po	packets out
oi	octets in
oo	octets out
bpi	broadcast packets in
bpo	broadcast packets out
pu	peak utilization
rbe	receive buffer errors
xbe	transmit buffer errors
fcs	frame check sequence errors
fa	frame alignment errors
c	segment collisions
rc	port collisions
tc	transmit collisions
q	output queue length
gp	giant packets
cu	current utilization
lc	local carrier
er	excessive retries
tm	table miss

Table 8.4 - Segment Level Statistic Parameters

Parameter	Description
ce	coding errors
drm	data rate mismatch
jbrs	jabbers
se	short events
rts	runts
lcs	rate collisions
ap	auto partitions
iss	isolates
sfc	source address field changes
lsa	source address of the last incoming packet.

<display restrictor> Specify a specific segment, range of segments or a comma-separated list of segments.

clear Clears all port and segment-level statistics.

The **`stats show`** command displays media-level statistics. If portstats are being collected, portstats statistics are displayed in place of segment-wide statistics unless the -s flag is used.

CHAPTER 9 NVRAM Subsystem

This chapter explains the NVRAM subsystem and the commands used to make changes in the NVRAM subsystem and booting parameters. The commands in the NVRAM subsystem affect the following PowerHub parameters:

- Boot source order
- PowerHub switch's IP address
- PowerHub local subnet mask
- Gateway address (when a gateway (router) separates the PowerHub from a BOOTP server)
- A TFTP server address
- Post-system crash behavior
- Segment allocation on NIM slots
- RIPv2 encryption key

9.1 NVRAM Configuration Commands

The NVRAM commands can be used to show, set, change or clear the parameters listed above. This section explains the commands used to set NVRAM variables using commands in the `nvram` subsystem. The commands in the `nvram` subsystem can be used to set the values used by the PowerHub for:

- Setting the boot order.
- Setting IP addresses to identify this PowerHub, default file server and gateway.
- Behavior following an unexpected system crash.
- The number of segments allocated to each NIM slot.

9.1.1 Boot Order

The boot order command **bo** is used to set a default PowerHub booting order. The syntax of the **bo** command is:

```
bo set <value> bo [show] bo unset
```

show|set|unset Sets, shows or unsets the boot order. The boot order designates the order of sources from which the PowerHub attempts to boot.

m|n The boot values can be the Compact Flash module (**m**) and/or network (**n**). The boot order can be set in any order.

If more than one boot source is specified, the PowerHub attempts them in the order specified. For example: if **mn** is entered, the PowerHub attempts to boot from Compact Flash and then the network.

NOTE

If more than one boot source is specified, the configuration files on each source should match to prevent an erroneous configuration from be loaded into the PowerHub.

The following examples show the results of the various boot order command options:

```
264:PHswitch:nvram# bo
bo              m  (flash-module)
265:PHswitch:nvram# bo set mn
266:PHswitch:nvram# bo
bo              mn  (flash-module,net)
267:PHswitch:nvram# bo unset
268:PHswitch:nvram# bo
bo              (not set, defaults to "f")
270:PHswitch:nvram# bo set m
```

9.1.2 My IP Address

The **myip** command is used to set the IP address of the PowerHub. The syntax of this command is:

```
myip [show] myip set <ipaddr> myip unset
```

show|set|unset Specifies whether to show, set, or unset (clear) the IP address associated with the PowerHub.

<ipaddress> Specifies the IP address of the PowerHub.

```
280:PHswitch:nvram# myip
myip            169.144.86.54
281:PHswitch:nvram# myip set 169.144.86.55
282:PHswitch:nvram# myip unset
```

9.1.3 My Subnet Mask

Use the **mysm** command to set the subnet mask of the PowerHub:

mysm [show] mysm set *<ipaddr-mask>* **mysm unset**

show|set|unset Specifies whether to show, set, or clear the IP subnet mask for the interface.

<ipaddr-mask> Specifies the PowerHub IP subnet mask.

```
285:PHswitch:nvram# mysm
mysm            255.255.255.0
286:PHswitch:nvram# mysm set 255.255.255.255
287:PHswitch:nvram# mysm unset
```

9.1.4 File Server IP Address

Use the **fsip** command to set the IP address of the file server:

fsip [show] fsip set *<ipaddr>* **fsip unset**

show|set|unset Specifies whether to show, set, or clear the IP address of the file server.

<ipaddress> A file server's IP address.

```
290:PHswitch:nvram# fsip
fsip            (not set)
291:PHswitch:nvram# fsip set 169.144.86.49
292:PHswitch:nvram# fsip unset
```

9.1.5 Gateway IP Address

Use the **gwip** command to set the IP address of the gateway server:

gwip [show] gwip set *<ipaddr>* **gwip unset**

show|set|unset Specifies whether to show, set, or clear the IP address of the gateway router.

<ipaddress> Specifies an intervening router's (gateway's) IP address.

The following examples display the use of these commands. In the first example, values are configured into the hub’s NVRAM to support “semi-prescient” netbooting.

```
295:PHswitch:nvram# gwip
gwip            (not set)
296:PHswitch:nvram# gwip set 169.144.86.49
297:PHswitch:nvram# gwip unset
```

9.1.6 Crash Reboot

The **crashreboot** command instructs the PowerHub to reboot automatically following a system crash. The syntax of this command is:

crashreboot [show] crashreboot set crashreboot unset

show|set|unset Specifies whether the switch automatically attempts a reboot following an unexpected system crash. The default is **set**, which causes the hub to attempt a reboot following a crash. We recommend that this setting not be changed unless instructed to do so by FORE Systems TAC.

```
302:PHswitch:nvram# crashreboot
crashreboot     (set)
303:PHswitch:nvram# crashreboot set
304:PHswitch:nvram# crashreboot unset
```

9.1.7 Slot Segments

The **slotsegs** command is used to allocate segments to specific slots. The syntax of this command is:

slotsegs [show] slotsegs[*<n>***] [show] slotsegs[***<n>***] set**
<segment-count> **slotsegs[***<n>***] unset**

<slot> Specifies the slot number for which segments are being allocated.

<num> Specifies the number of segments being allocated to the specified slot.

NOTE The brackets around the slot number must be entered as they are part of the command.

In the following example, segments are allocated to some NIM slots:

- Four segments are allocated on NIM slot 1 (enough for a 4x4 or 4x6 Microsegment Ethernet Module).
- Two segments are allocated on NIM slot 4 (enough for a Dual FDDI Module).
- 16 segments are allocated on NIM slot 3 (enough for a 16x1 Ethernet Module).

```
307:PHswitch:nvram# slotsegs
slotsegs[ 1]    16
slotsegs[ 2]    16
slotsegs[ 3]    2
slotsegs[ 4]    (not set)
slotsegs[ 5]    (not set)
slotsegs[ 6]    (not set)
slotsegs[ 7]    (not set)
slotsegs[ 8]    (not set)
slotsegs[ 9]    (not set)
slotsegs[10]    (not set)
slotsegs[11]    (not set)
slotsegs[12]    (not set)
slotsegs[13]    (not set)
slotsegs[14]    (not set)
slotsegs[15]    (not set)
slotsegs[16]    (not set)
slotsegs[17]    (not set)
slotsegs[18]    (not set)
slotsegs[19]    (not set)
slotsegs[20]    (not set)
................Total segments reserved: 34
308:PHswitch:nvram# slotsegs[3]
slotsegs[3]    2
................Total segments reserved: 34
309:PHswitch:nvram# slotsegs[1] set 32
310:PHswitch:nvram# slotsegs[1]
slotsegs[1]    32
................Total segments reserved: 50
312:PHswitch:nvram# slotsegs[1] unset
```

9.2 RIPv2 Authentication

RIPv2 supports encrypted packet transmission using the MD5 algorithm to authenticate route and table updates. The MD5 algorithm allows packets to be encrypted at a source PowerHub and decoded at a destination PowerHub containing the same encryption key and key-string (password). Because the keyID is not transmitted over the network, but is set at each end, it reduces the likelihood of a successful attack on the network.

MD5 authentication is only supported in RIPv2. It does not work in RIPv1. RIPv2 must be enabled and running on all interfaces that require authentication. Additionally, RIPv2 authentication is not supported on interfaces that are configured for both RIPv1 and RIPv2; interfaces must be configured for RIPv2 only.

The MD5 key must be set up on the PowerHubs at both sides of the connected interfaces in order for the authentication to take place. The keyid and the key-string must be the same on both PowerHubs. Refer to RFC-2082 for a discussion on RIPv2 authentication using the MD5 encryption algorithm. The syntax for the **md5key** command is:

md5key [show] md5key[*<keyid>***] [show] md5key[***<keyid>***]**
set *<key-string>* **md5key[***<keyid>***] unset**

[keyid] Specifies the number or identifier of the MD5key. The number must be a whole number between 1 and 255 The brackets around the keyid are part of the command and must be included. This variable is not required with the **show** argument.

<key-string> Specifies the password for the encryption. The maximum password length is 16 characters. This variable is only required with the **set** argument.

Example of the **md5key** command are contained below:

```
338:PHswitch:nvram# md5key
Total keys reserved: 0
339:PHswitch:nvram# md5key[1] set abcdefghijklmnopq
Error:  Authentication string is too long: abcdefghijklmnopq
340:PHswitch:nvram# md5key[1] set abcdefghijklmnop
341:PHswitch:nvram# md5key
md5key[1]      (set)
..................Total keys reserved: 1
342:PHswitch:nvram# md5key[1] unset
343:PHswitch:nvram# md5key
..................Total keys reserved: 0
```

In this example, key 1 is set with the keyID (password) of “abcdefghijklmnop.” This is the only time the keyID displayed.

CHAPTER 10 Host Commands

This chapter describes the commands in the `host` subsystem and tells you how to use these commands to perform the following tasks:

- Display the TCP configuration settings.
- Display the TCP table.
- Display TCP, TELNET, and UDP statistics.
- Clear TCP, TELNET, and UDP statistics.
- Set the connection time.
- Set the keep-alive interval.
- Kill a TCP connection.
- Display the UDP table.

The PowerHub software's `host` subsystem includes an implementation of the TCP (Transmission Control Protocol) stack, a connection-oriented, industry-standard protocol for moving data between nodes in a network environment. In particular, TCP is used by TELNET, a program that allows workstations to communicate with the hub using an in-band network connection. To define TCP filters for controlling access to your network, refer to the *PowerHub Filters Manual*.

10.1 Accessing the Host Subsystem

To access the `host` subsystem, issue the following command at the runtime command prompt:

```
host
```

To list available commands in the `host` subsystem issue **`help`** or **`?`**.

10.2 Displaying the TCP Configuration

Use the **`config [show] tcp`** command to display configuration parameters used by the `host` subsystem.

```
67:PowerHub:host# config show tcp
TCP Configuration
-----------------

Round Trip Algorithm:                   vanj
Min Rexmit Interval:                    1000 ms
Max Rexmit Interval:                   64000 ms
Max Connections Allowed:                   2

connection-idle-time:                     20 minutes
keep-alive-interval [kainterval]:         75 seconds
keep-alive delay [kadelay]:            1200 seconds
Time to disconnect on idle conn:          30 minutes 0 seconds
```

This display shows the following information about the current TCP configuration parameters:

- The round-trip algorithm used by the PowerHub software is the Van Jacobson algorithm.
- The minimum retransmit interval is 1,000 milliseconds.
- The maximum retransmit interval is 64,000 milliseconds.
- The maximum number of simultaneous TELNET (TCP) connections to the PowerHub system that can be supported is two.
- The connection-idle time is 20 minutes.
- The keep-alive interval is 75 seconds.
- The keep-alive delay is 1200 seconds.
- The time allowed before an idle connection is automatically disconnected is 30 minutes. This value is based on the values of the connection-idle time and the keep-alive interval.

10.3 Displaying the TCP Table

The TCP table lists the active TCP connections between the hub and other devices. Use the **`status show tcp`** command to list the TCP table.

Following is an example of the TCP table:

```
7:PowerHub:host# status show tcp
                     Active TCP Connections
Conn Id  Rem IP Addr     Rem Port  Loc IP Addr      Loc Port Conn. State
-------  -------------   --------  --------------   -------- -----------
16       147.128.128.128  1043     147.128.128.64   23       ESTABLISHED  **
17       147.128.128.8    1201     147.128.128.64   23       ESTABLISHED
List of registered UDP clients:
161 SNMP
520 RIP
```

For each TCP connection to the hub, the TCP table shows information under the following headings:

Conn ID A unique integer that identifies the connection. This identifier can be used to terminate the connection using the **`kill <connection-id>`** command.

Rem IP Addr The IP address of the remote device that initiated the connection.

Rem Port A process port number for the remote device (management station). Note that the process port number is unrelated to the PowerHub physical port or segment numbers. It is assigned by the remote operating system.

Loc IP Addr The IP address of the local device. This is always the hub.

Loc Port A process port number. Unrelated to the PowerHub physical port or segment numbers. It is a “well-known” port number used by the TELNET process.

Conn. State The connection state is one of the following states of the standard TCP software state machine:

CLOSED **CLOSING**
CLOSE-WAIT **ESTABLISHED**
FIN-WAIT-1 **FIN-WAIT-2**
LAST-ACK **LISTEN**
SYN-RECEIVED **SYN-SENT**
TIME-WAIT

Most of these states are never displayed by the **`status show tcp`** command because they occur for a short time. Connections in the CLOSED or LISTEN state are not displayed.

The current TELNET session (if you are connected through TELNET) is indicated by two asterisks (**) following the table entry for that session.

10.4 Displaying Statistics

The `host` subsystem maintains statistics on TCP, TELNET, and UDP packets. TCP and UDP statistics are a superset of the corresponding statistics provided in the SNMP MIB. (There is no TELNET MIB.) The software maintains two copies of each TCP, TELNET, and UDP statistics counter:

- Count since last statistics clear.
- Count since last system reset.

Use the **`stats`** command to display statistics. Here is the syntax for this command:

```
stats [show|clear] [-i] [-t] tcp|tel[net]|udp
```

tcp|telnet|udp Specifies the type of protocol for which you want to display packet statistics.

-t Displays statistics totals collected since the last system reset, rather than the statistics collected since the last statistics clear.

In the following example, TCP statistics collected since the last statistics clear are displayed.

```
62:PowerHub:host# stats tcp
TCP Connection & Pkt statistics (count since last stats clear):
Active Opens:                      0
Passive Opens:                     5
Failed Conn Attempts:              0
Resets In Estb State:              0
Current Open Conns:                2
Segments Received:              1088
Segments Sent:                  1030
Rexmitted segments:                2
Segments Rcvd With Err:            0
Resets Sent:                       0
Short Segments Rcvd:               0
```

10.4.1 Clearing Statistics

Use the **`stats clear`** command to clear statistics for TCP, TELNET, or UDP packets. Here is the syntax for this command:

```
stats clear [-i] [-t] tcp|tel[net]|udp
```

tcp|tel[net]|udp Specifies the type of protocol for which you want to clear packet statistics.

When you clear statistics, the counters that record statistics since the last clear are reset to zero. The PowerHub software immediately begins collecting new statistics. The counters that record statistics since the last system reset are unaffected by the **`stats clear`** command.

10.5 Setting TCP Session Parameters

The PowerHub software can kill idle TCP (TELNET) connections automatically using the following TCP configuration parameters:

- Connection-idle time (default 20 minutes).
- Keep-alive interval (default 75 seconds).

The combination of the two parameters above determines the time allowed to pass before an idle connection is automatically disconnected.

10.5.1 The Connection Idle Time and Keep Alive Interval

When a new TELNET connection is established, an idle timer is started. The idle timer is reset to 0 and restarted whenever there is activity on the connection. If the idle timer reaches a preset value, the *connection idle time*, then the hub sends a keep-alive packet to the remote device. If there is still no activity, the hub continues to send keep-alive packets at an interval called the *keep-alive interval*, until eight keep-alive packets are sent. If there is still no activity, then the connection is dropped.

A connection is automatically dropped if it is idle for a period of time equal to the connection-idle time plus eight times the keep-alive interval. Using the default values of these parameters, the maximum idle time is 30 minutes.

In addition, you can display and set the control characters used for displaying and editing text during a TELNET session.

10.5.1.1 Setting the Connection Idle Time

Use the **set kadelay** command to specify how long a TELNET connection can remain idle before sending keep-alive packets. The syntax for this command is:

```
set kadelay|kad <time>
```

<time> Specifies how many minutes to allow a TCP (TELNET) connection to remain idle before sending keep-alive packets. The range is **5** to **30** minutes; the default is **20** minutes.

10.5.1.2 Setting the Keep-alive Interval

Use the **set kainterval** command to specify how often the hub sends keep-alive packets before ending a connection. Here is the syntax for this command:

```
set kainterval|kai <time>
```

<time> Specifies how often to send keep-alive packets before ending a connection. The range is **30** to **240** seconds; the default is **75** seconds.

10.5.2 Closing a TCP Connection

At any time, you can end a TCP (TELNET) connection. The command you use depends upon whether you are ending the:

- Connection from which you are working.
- Connection other than the one from which you are working.

10.5.2.1 Current TCP Connection

Use the **logout** command to end the current TCP connection.

10.5.2.2 Another TCP Connection

Use the **kill** command to end a TCP connection other than the one in which you are working. You must issue this command from a session other than the one you are ending. The syntax for this command is:

```
kill <connection-id>
```

<connection-id> Specifies the ID assigned to the session by the hub when the session was established. To determine what the connection ID is, use the **stats tcp** command to display the TCP table. The connection IDs are listed in the first column, under Conn ID.

10.6 Displaying the UDP Table

The PowerHub software contains agents that can respond to certain "well-known" UDP port requests, such as RIP packets and SNMP requests. The ports listed in the UDP port table are the port numbers that UDP clients register with the UDP protocol code.

When the hub receives a UDP packet, it checks the UDP port number specified in the packet against the list of UDP ports in the UDP port table. If the PowerHub can respond to the UDP request, it does so. If the PowerHub cannot respond to the UDP request, it does one of the following:

- Drops the packet.
- Forwards the packet to the device specified by the IP Helper address, if an IP Helper address has been configured for the segment on which the UDP packet is received. See *Chapter 15, Configuring IP Routing* for information on using the PowerHub IP Helper feature.

To display a complete list of the UDP protocol ports supported by the PowerHub software, issue the **status udp** command. The numbers and names are "well-known" UDP protocol port numbers and names as defined in RFC 1700.

Following is an example of the display produced by this command:

```
81:PowerHub:host# status udp
List of registered UDP clients:
 161 SNMP
 520 RIP
 67  BOOTPS
 68  BOOTPC
```

The UDP ports listed in this display indicate that the PowerHub contains agents for processing UDP packets sent to UDP protocol ports 161, 520, 67, and 68. In other words, the PowerHub system supports the following types of UDP packets:

- SNMP
- IP RIP
- BOOTP (on server side)
- BOOTP (on client side)

NOTE

For information on using bridge filters and templates, see the *PowerHub Filters Reference Manual*.

CHAPTER 11 FDDI Commands

This chapter explains the commands used to display, configure, and adjust parameters related to the FDDI connections. This chapter describes the following functions available through the fddi subsystem:

- Attaching FDDI concentrators to an FDDI segment to enable a DAC (Dual Attached Concentrator).
- Detaching an FDDI concentrator from PowerHub segments.
- Adjusting hardware timer.
- Displaying the configuration of the PowerHub concentrator configuration.
- Displaying the current FDDI-specific statistics.
- Displaying the values of a specified group of FDDI MIB objects.

11.1 Accessing the FDDI subsystem

To access the commands in the **`fddi`** subsystem, issue the following command from any PowerHub runtime command prompt:

```
fddi
```

11.2 Attaching a FDDI Concentrator

Use **`concentrator attach|detach`** command to attach, or detach FDDI Concentrator modules to a single FDDI segment. Up to four FDDI Concentrator modules can be attached to a single FDDI segment, for a total of up to 64 FDDI Concentrator ports. The second FDDI concentrator channel can be used to attach up to four concentrators to an additional FDDI segment. Up to two DAS segments can be configured as DACs, provided they are in the same FDDI module. Note that a single FDDI Concentrator module cannot be attached to two DAS segments (DACs). See the *PowerHub Hardware Reference Manual* for information about the FDDI Concentrator Modules.

NOTE

The **attach concentrator** command resets the FDDI module to attach the A and B port of the FDDI segment and the M ports of the FDDI Concentrator to the same ring. This reset causes a slight delay of approximately 10 seconds while the ring is reconfigured to add the newly attached FDDI Concentrator module.

The syntax of this command is:

```
concentrator|con attach|detach [<slot-list> to <fddi-segment>]
concentrator|con [show]
```

attach|detach Attach or detach a concentrator to the FDDI segment.

<slot-list> Specifies the slots that contain the Concentrator modules to attach to the specified FDDI segment. Up to three slot numbers can be specified. Separate the slot numbers with spaces.

<fddi-seg> Specifies the FDDI segment the FDDI Concentrator Module is to be attached.

[show] Displays the current FDDI configuration.

In the following example, the FDDI Concentrator modules in NIM slots 4 and 5 are being attached to FDDI segment 32. This command configures FDDI segment 32 as a DAC.

```
4:PowerHub:fddi# concentrator attach 4 5 to 32
```

If additional Concentrator modules are being attached to a DAC, use the **attach** option to add the module. In the following example, the Concentrator module in NIM slot 6 is attached to the DAC in segment 32, which is already managing the Concentrator module in NIM slots 4 and 5.

```
5:PowerHub:fddi# concentrator attach 6 to 32
```

To display the current FDDI Concentrator configuration, issue the following command:

```
show concentrator
```

In the following example, the FDDI Concentrator configuration created by the **concentrator attach** commands shown in the previous examples is displayed.

```
6:PowerHub:fddi# concentrator
concentrator cards are in slots: 4 5 6
concentrator configuration:
DAC            Concentrator Module
32              4 5 6
```

11.3 Verifying Attachment to the FDDI Segment

The **smtmib** command is used to display the FDDI Concentrator MIB variables for specified FDDI port(s). The syntax for this command is:

```
smtmib [show] [<group>] [<disprestrict>]
```

<group> Can be one of the following:

- **`smt`** Displays the FDDI MIB objects in the smt group.
- **`mac`** Displays the FDDI MIB objects in the MAC group.
- **`port`** Displays the FDDI MIB objects in the Port group.
- **`all`** Displays the FDDI MIB objects in all groups.

<disprestrict> Specifies the FDDI segment(s) to display the MIB objects. A single segment, a comma-separated list of segments, or a hyphen-separated list of segments can be specified.

An example of the display produced by this command.

```
1:PowerHub:fddi# show smtmib 8 port
PORT variables of port# 8
----------------------------
FDDI Port 1:
Port PC Type:          A-port
Port PC Neighbor:      None
FDDI Port 2:
Port PC Type:          B-port
Port PC Neighbor:      None
FDDI Port 3:
Port PC Type:          M-port
Port PC Neighbor:      None
FDDI Port 4:
Port PC Type:          M-port
Port PC Neighbor:      S-port
FDDI Port 5:
Port PC Type:          M-port
Port PC Neighbor:      S-port
FDDI Port 6:
Port PC Type:          M-port
Port PC Neighbor:      S-port
FDDI Port 7:
Port PC Type:          M-port
Port PC Neighbor:      S-port
FDDI Port 8:
Port PC Type:          M-port
Port PC Neighbor:      S-port
```

FDDI Port Specifies the FDDI port number within the displayed segment or DAC. These ports correspond to ports A and B. The A and B ports of the segment or DAC are listed as ports 1 and 2. For FDDI segments that are not attached to FDDI Concentrator modules, only two ports are listed.

For DACs, the A and B ports of the DAC are listed and the ports on the FDDI Concentrator module(s) attached to the DAC also are listed. The FDDI Concentrator ports are numbered from left to right. If more than one FDDI Concentrator module is attached to the DAC, the ports are listed from left to right, top to bottom.

Port PC Type Shows the FDDI port type. Port 1 is always A and port 2 is always B. FDDI Concentrator ports are always M.

Port PC Neighbor Shows the FDDI port type at the other end of the FDDI cable. Valid values are A, B, M, S, or none. The value displayed shows the most recent change to the port type. The displayed value does not change until the port is attached to another port type. Accordingly, if the FDDI cable is unplugged, the port type to which it was attached is still displayed. The value is "none" only if no device has been attached to the port.

In the above example, a 1x6 FDDI Concentrator Module has been attached to FDDI segment 8. The FDDI ports labeled 1 and 2 are the A and B ports on the DAC (the FDDI segment attached to the FDDI Concentrator module). The ports numbered 3 through 8 are the M ports on the Concentrator module attached to the DAC. Notice that for ports 1, 2, and 3 the Port PC Neighbor is listed as "None." This indicates that these ports are not connected by FDDI cables to other FDDI devices. Ports 4 through 8 are connected to S ports.

11.4 Setting FDDI Hardware Timers

There are two FDDI Concentrator hardware timers that can be set from the **fddi** subsystem. These timers are Target Token Rotation Time (TTRT), specified with the **treq** command, and Valid Transmission Time (TVX), specified with the **tvx** command. The following paragraphs describe these timers, the syntax of the commands, and the parameters available in the use of these commands.

11.4.1 treq Command

The **treq** command sets the target token rotation time (TTRT) variable (T_REQ). The TTRT specifies the amount of time each FDDI station holds on to the FDDI token. If the T_REQ timer for a segment is adjusted, it can change the amount of time each device attached to the FDDI segment holds on to the token. The syntax for this command is:

```
treq pset <time>|default <portlist>
```

<time>|default Specifies, in milliseconds, the new value for the hardware timer. The time must be followed by an "m" to indicate milliseconds.

Enter 4 to 167 milliseconds. An integer value must be specified; decimal numbers are truncated after the decimal. The default is **167**.

If **`default`** is specified, the timer resets to the default value.

<portlist> Specifies the FDDI segments for which to adjust a hardware timer.

Examples of the use of the **`treq`** command is shown below:

```
20:PHswitch:fddi# treq pset 25m 3.1
Segment 3.1 treq set to 25 millisecond
23:PHswitch:fddi# treq pset default 3.1
Segment 3.1 treq set to 167 milliseconds
```

11.4.2 tvx Command

The **tvx** command sets the valid time transmission variable (TVX). The syntax for this command is:

`tvx pset` *`<time>`***`|default`** *`<portlist>`*

<time>|default Specifies the new value for the hardware timer. The time entered must be followed by a "u" to indicate microseconds or an "m" to represent milliseconds.

If adjusting the TVX timer, specify from 2621 - 5200 microseconds (2.6 to 5.2 milliseconds). A number expressed up to three decimal places may be specified. Numbers with more than three decimal places are truncated after the third decimal. The default is **2621**.

<portlist> Specifies the FDDI segmen(t)s for which to adjust a hardware timer.

Examples of the use of the **`tvx`** command is shown below:

```
47:PHswitch:fddi# tvx pset 4700u 3.1
Segment 3.1 tvx set to 4.00 milliseconds
48:PHswitch:fddi# tvx pset 4m 3.1
Segment 3.1 tvx set to 4.00 milliseconds
51:PHswitch:fddi# tvx pset default 3.1
Segment 3.1 tvx set to 2.00 milliseconds
```

Generally, these timers do not need to be adjusted. However, if it is necessary to adjust them, use the commands described in the following sections to do so.

11.5 Displaying FDDI Statistics

Statistics on FDDI modules can be displayed using the **`status`** command. This command displays information that is tracked with the FDDI counters.

NOTE

Statistics tracked with the **`status`** command are not specific to a FDDI segment, but the module as a whole.

In addition to the standard Ethernet and FDDI packet statistics available using the **`bridge stats`** command (see *Chapter 13, Bridge Commands* for information on the **`bridge stats`** command) statistics that apply specifically to the FDDI modules are displayed. For each FDDI module, the software maintains counters for the following FDDI packet statistics:

- Number of packets forwarded locally (from one FDDI segment to the other) on the module. This number is always 0 for the Single FDDI module and the Universal Single FDDI module.
- Number of packets forwarded to the Packet Engine because the FDDI Engine could not make a forwarding decision for the packet.
- Number of packets locally filtered by the FDDI module.
- Number of fragmented packets.
- Number of un-fragmented packets.
- Number of large IP packets fragmented by the FDDI module. A large IP packet is one that exceeds the Ethernet MTU (maximum transmission unit) size.
- Number of large IP packets that needed to be fragmented, but could not be fragmented, and therefore were dropped by the FDDI module. This can occur if the packet’s No Fragment bit is set, or if the switching engine temporarily runs out of resources for fragmenting packets.

NOTE

Statistics for specific segments are available using the **`show smtmib`** *<seg-list>* **`priv`** command. These statistics do not apply to the FDDI Concentrator modules.

To display the packet statistics for a module, issue the following command:

```
status [show] counter <slot>
```

<slot> Specifies the slot in which the FDDI module is installed. If the slot number is not known, use the **`system config`** command to display the slot locations of the FDDI modules. Alternatively, visually check the slot-number label, located to the left of the modules in the PowerHub chassis.

An example of the information displayed by this command.

```
154:PHswitch:fddi# status counter 3
FDDI Counters of slot# 3:
---------------------------
Number of Packets Forwarded to FDDI             0
Number of Packets Forwarded to Packet Engine    0
Number of Packets Filtered                      0
Number of Packets Fragmented                    0
Number of Packets Not-Fragmented                0
Number of IP Packets Forwarded Locally          0
Number of IP Packets Dropped (BAD CHKSUM)       0
155:PHswitch:fddi#
```

11.5.1 Displaying FDDI Information

The following commands provide information on the state of FDDI Concentrators present in the PowerHub. These commands give information on the DAC, reset count for the PowerHub, and nvram in the concentrators. To display information about the FDDI DAC, enter the following command:

`[show] dac`

```
159:PHswitch:fddi# dac
There is no DAC configured
```

To display the FDDI DAC NVRAM, type the following command:

`[show] nvram`

```
161:PHswitch:fddi# src
FDDI Reset Count:
        PORT 33:        0
```

To display the FDDI reset count for all FDDI port(s), type the following command:

[show] resetct|src

```
163:PHswitch:fddi# show nvram
slot[1]:
slot[2]:
slot[3]:
slot[4]:
slot[5]:
slot[6]:
slot[7]:
slot[8]:
slot[9]:
slot[10]:
slot[11]:
slot[12]:
slot[13]:
164:PHswitch:fddi#
```

CHAPTER 12 TFTP Commands

The `tftp` subsystem contains the PowerHub implementation of TFTP (Trivial File-Transfer Protocol). Use the `tftp` subsystem commands to perform the following tasks:

- Set the default TFTP server.
- Display the default TFTP server.
- “Unset” the default TFTP server.
- Download or display a file stored on a TFTP server.
- Upload a file from the hub’s floppy drive or Flash Memory Module to a TFTP server.
- Load (activate) a configuration file stored on a TFTP server.
- Save PowerHub configuration changes to a configuration file on a TFTP server.

To use the commands in this subsystem, you must configure your TFTP server to support TFTP file transfers. The procedures for configuring your server depend upon the particular type of server you are using. See your server documentation for configuration information.

Also, the PowerHub segment that connects the hub to the TFTP server must have an IP interface defined on it. For information about adding an IP interface, see *Chapter 15, Configuring IP Routing*.

NOTE

The TFTP protocol provides no authentication for any services, including downloading or changing files stored on the TFTP server. If you configure your TFTP server to allow the `tftp` commands to be used, anyone with access to the server can download or change files.

12.1 Accessing the TFTP Subsystem

To access the `tftp` subsystem, issue the following command at the runtime command prompt:

```
tftp
```

12.2 Considerations

The PowerHub TFTP commands work with many types of TFTP servers, including servers running UNIX, DOS, Windows, OS/2, or other operating systems. The following considerations apply to TFTP servers that are running UNIX, a very common platform for TFTP.

Regardless of the platform used for the TFTP servers in your network, we recommend that you consult your server's documentation regarding:

- File permissions (not applicable to some operating systems).
- Conventions for pathnames and file names.

If you experience problems uploading or downloading files between the PowerHub switch and your TFTP server, you often can resolve the problems by verifying whether the switch has or needs read and write access to the server, and how file names need to be specified to the switch.

12.2.1 TFTP Commands and UNIX Read/Write Permissions

To use the PowerHub TFTP commands to upload or download a file, or to save or read a PowerHub configuration file, the proper UNIX read/write permissions must be set on the TFTP server. On most servers, permissions are controlled separately for users, groups, and "others." The TFTP server considers the PowerHub system to be among the "others."

You can control read/write access to PowerHub files and directories on the TFTP server by setting the read and write permissions. On most UNIX systems, you can display permissions information using the UNIX **ls** command. Here is an example of the permissions information displayed for a file on a TFTP server. The display on your TFTP server might be different.

```
$ ls -l
total 3
-rw-rw---- 1 mrspat     622 Jul 19 15:09  Lab1.env
-rw-rw-r-- 1 ethan      643 Jul 19 15:11  Lab2.env
-rw-rw--w- 1 sascha     611 Jul 19 15:13  Lab3.env
-rw-rw-rw- 1 stripie    698 Jul 19 15:15  Lab4.env
-rw-rw-rw- 1 tiger      698 Jul 19 15:15  Lab5.env
```

The text shown in bold is the permission information for each file, for "others."

- In this example, no read or write permissions are enabled for others for `Lab1.env`. Consequently, you cannot download this file or upload a file by this name using the PowerHub TFTP commands.
- Read permission, but not write permission, is granted to the file `Lab2.env` for others. You can use the PowerHub TFTP commands to display or download this file, but you cannot upload a file by this name.

- The file Lab3.env cannot be downloaded using the PowerHub TFTP commands. You can, however, upload a file by this name.
- Finally, both read and write permissions are enabled for the file Lab4.env and Lab5.env. You can download these files and upload a file by these names.

On most TFTP servers, you can change permissions using the UNIX **chmod** command. See the documentation for your UNIX shell for details.

12.2.2 PathNames

Depending upon your TFTP server configuration, you might need to specify pathnames with the TFTP commands.

On some TFTP servers, when you use the PowerHub TFTP commands to upload or download files, the PowerHub software understands file names according to where the PowerHub system accesses the server. You can upload or download only those files that are located in the directory where the PowerHub accesses the server, or in one of that directory's subdirectories. Also, if the file is in a subdirectory, you must specify the pathname along with the file name.

For example, suppose you configure your TFTP server to allow the PowerHub system to access the server at a directory called TFTP.

```
TFTP
       fore
                  ph
                      ethan.env
                      sascha.env
```

All directories below the TFTP directory are considered part of the pathname for the files stored there. Relative to the PowerHub switch, the pathname for the files ethan.env and sascha.env is fore/ph. To download the file ethan.env, you would issue the following command:

```
get -a fore/ph/ethan.env ethan.env
```

-a Specifies net-ASCII mode. (Files are transferred in binary mode by default.

fore/ph/ethan.env Is the file name, including the pathname.

ethan.env Is the name you want the file to have on the PowerHub system. This name must be in DOS format (filename.ext).

12.2.3 File Naming Conventions

Local file names are optional with the PowerHub **`tftp get`** command. You can omit the local file name if the file is not in a subdirectory and the file name is eight characters or fewer in length with an extension no longer than three characters.

Suppose the PowerHub system has access to the TFTP server at the TFTP directory, as in the following example:

```
TFTP
    fore
        ph
            sascha.env
            ethan.env
            lotsofdots
```

If the **`get`** command is issued without specifying the local file name (**`sascha.env`**), an error message is displayed on the PowerHub terminal.

In addition, the PowerHub system uses DOS file-naming conventions, but the file lotsofdots does not fit the DOS file-naming conventions. To download lotsofdots, you need to specify a local file name that fits the DOS file naming conventions, as in the following example:

```
get -a fore/ph/lotsofdots spots
```

In this example, the file lotsofdots is named spots on the PowerHub system.

12.2.4 Remote File Names

Some TFTP servers require that the remote file name exist on the server before you can write to that file name. If your server requires that the file name already exist, create a zero-length file on the server, then specify the name of that file as the remote file name with the **`put`** or **`savecfg`** commands.

Also, on some TFTP servers, files that you overwrite on the server are not properly truncated. When you overwrite an existing file on the TFTP server, if the older version of the file is longer than the new file, the older version is not truncated properly by the server. As a result, the new version of the file contains part of the older version of the file. If you are unsure whether the new version will completely replace the older version of a file, do one of the following:

- Remove the older version of the file, then save the new version.
- If your server requires that the file name be present on the server before you can copy to it, create a zero-length file under a new name, then save the PowerHub file under the new name. After the new file is copied to the server, delete the older version of the file and rename the new file as desired.

12.3 Setting, Displaying, or Unsetting the Default Server

The commands described in the following sections let you specify a particular TFTP server to be used in the file operations described in this chapter.

If you choose not to specify a default TFTP server, you still can specify a server with the individual TFTP commands.

12.3.1 Setting the Default TFTP Server

Use the **`set`** command to specify the default TFTP server. Here is the syntax for this command:

```
set server <ipaddress>
```

<ipaddress> Specifies the IP address of the TFTP server you want to use as the default. Specify the address in dotted-decimal notation.

You can only have one active TFTP server at a time. Setting a new default TFTP server's IP address replaces the existing TFTP server's IP address.

12.3.2 Displaying the Default TFTP Server

Use the **`show`** command to display the IP address of the default TFTP server. Here is the syntax for this command:

```
show server
```

The show server command shows you the TFTP server at the IP address you specify with the **`set server`** command.

12.3.3 Removing the Default TFTP Server Setting

Use the **`unset`** command to remove the default TFTP server setting. Here is the syntax for this command:

```
unset server
```

12.4 Downloading or Displaying a File

Use the `get` command to display or download a file stored on a server. Here is the syntax for this command:

```
get [-h <host>] [-a] <remote-file> [<local-file>|tty]
```

-h <host> Specifies the IP address, in dotted-decimal notation, of the TFTP server. If you do not specify this argument, the default server is used. The default server is specified using the `set server` command. (See Section 12.3.1**.**)

-a Forces the transfer to take place in net-ASCII transfer mode, rather than octet mode. Octet mode transfers the file, including end-of-line characters, exactly as it is stored on the server. Net-ASCII changes the end-of-line characters to be compatible with the display or storage device that receives the file.

Use the default (octet-mode) to download software image files (ex: 7f, 7pe, 7atm, and so on). Use the net-ASCII mode to download configuration files, environment files, and other text files.

If you plan to display the file on your management terminal (by specifying `tty` as the local file name), omit this argument. The file is automatically transferred in net-ASCII format.

<remote-file> Specifies the name of the remote file. Specify the name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `transfer` and this directory is specified as the TFTP home directory, do not specify `transfer` as part of the file name.

<local-file>|tty If you omit this argument, the PowerHub software assumes that you want to use the same file name on the server and include the pathname (if any) in the file name.

If you omit the *`<local-file>`* argument or specify a local file name, the file is written to a local storage device.

NOTE

If you are using a PowerHub 6000, the file is written to the Flash Memory Module (local storage device). If you are using a PowerHub 7000, the file is written to the device from which the hub was booted[1]. To specify the other device (not the default), preface the file name with **fm**: (Flash Memory Module) or **fd**: (floppy diskette). If the software was booted over the network, the floppy drive is the default device (local storage device).

If the file name on the server is an invalid pathname on the PowerHub, an error message is displayed on the PowerHub terminal.

If you specify **tty**, the file is not downloaded to your system, but an image of the file is displayed on the management terminal. You can display the file from within a TTY (RS-232) session or a TELNET session.

If you do not specify a TFTP server name and no default server name has been configured, an error message is displayed. To configure a default server name, use the **set server** command. (See Section 12.3.1.)

12.5 Uploading a File

Use the **put** command to upload a file stored on the floppy diskette or Flash Memory Module of a PowerHub to a TFTP server. Here is the syntax for this command:

put [-h *<host>*] [-a] *<localfile>* [*<remote-file>*]

-h <host> Specifies the IP address, in dotted-decimal notation, of the TFTP server. If you do not specify this argument, the default TFTP server is used. The default TFTP server is specified using the **set server** command. (See Section 12.3.1.)

[1] To determine the device from which your hub was booted, issue the **system bootinfo** command. (See the `System` subsystem chapter in the PowerHub Hardware Reference Manual for your PowerHub switch.)

-a Forces the transfer to take place in net-ASCII transfer mode, rather than octet mode. Octet mode transfers the file, including end-of-line characters, exactly as it is stored on the server. Net-ASCII changes the end-of-line characters to be compatible with the display or storage device that receives the file.

Use the default (octet-mode) to download software image files (ex: 7f, 7PE, ppu. 7PE, and so on). Use the net-ASCII mode to download configuration files, environment files, and other text files.

<local-file> Specifies the local file name.

NOTE

If you are using a PowerHub 6000, the files are present on the Flash Memory Module. If you are using a PowerHub 7000, the file is written to the device from which the hub was booted[1]. To specify the other device (not the default), preface the file name with **`fm`**: (Flash Memory Module) or **`fd`**: (floppy diskette). If the software was booted over the network, the floppy drive is the default device.

<remote-file> Specifies the name of the file as you want it to appear on the server. Specify the name that is meaningful to the TFTP program on the server. For example, if the name with the path of the server contains a subdirectory called `transfer` and this directory is specified as the TFTP home directory, do not specify transfer as part of the file name.

In the following example, an environment file is uploaded to a UNIX server.

```
12:PowerHub:tftp# put -a /fore/env/hub1.env
177.177.45.20:/fore/env: 833 bytes
13:PowerHub:tftp#
Notice that a pathname is specified with the file name in this example. Make sure you
specify the pathname that is meaningful to your TFTP program.
```

[1] To determine the device from which your hub was booted, issue the **`system bootinfo`** command. (See the `System` subsystem chapter in the PowerHub Hardware Reference Manual for your PowerHub switch.)

On UNIX machines, if the write permission for "others" is not enabled on the TFTP server for the file name or the directory to which you are trying to write the file, a message such as the following is displayed:

```
12:PowerHub:tftp# put -a hub1.env
tftpWrite: Peer generated error
tftp: Permission denied: Access violation
13:PowerHub:tftp#
If you receive this error, check the file and directory permissions for "others" on the
TFTP server.
```

If your TFTP program is on a UNIX machine and that machine requires that the file name already exist, but the file does not yet exist on the server, a message such as the following is displayed on the PowerHub terminal:

```
14:PowerHub:tftp# put hub2.env
tftpWrite: Peer generated error
tftp: File not found: File not found
15:PowerHub:tftp#
```

12.6 Loading a Configuration File

During normal run-time operation of the hub, you can read (load) a configuration file stored on a remote TFTP server. To do so, issue the following command:

readcfg|rdcfg [-v] [-h *<host>*] *<remote-file>*

-v Displays commands to the open TELNET session as they are executed.

-h <host> Specifies the IP address of the TFTP server. If you do not specify this argument, the default TFTP server is used. (The default TFTP server is specified using the **set server** command. SeeSection 12.3.1.

<remote-file> Specifies the name of the configuration file you want to load. Specify the name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called configs and this directory is specified as the TFTP home directory, do not specify **configs** as part of the file name.

As with the **get** command, if you do not specify a host server name and no default server name has been configured, an error message is displayed.

12.7 Saving a Configuration File

During normal run-time operation of the hub, you can save the hub's current configuration to a file on a remote TFTP server. To save configuration files, issue the following command:

savecfg [-h *<host>*] *<remote-file>*

-h <host> Specifies the IP address of the TFTP server. (The default server is specified using the **set server** command. See Section 12.3.1

<remote-file> Specifies the name of the configuration file you want to save. Specify the name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `configs` and this directory is specified as the TFTP home directory, do not specify **configs** as part of the file name.

On UNIX-based TFTP servers, if the write permission for "others" is not enabled for the configuration file name or the directory to which you are trying to write the file, a message such as the following is displayed:

```
16:PowerHub:tftp# savecfg ace.cfg
tftpWrite: Peer generated error
tftp: Permission denied: Access violation
17:PowerHub:tftp#
```

If you receive this error, check the file and directory permissions for "others" on the TFTP server.

If your UNIX-based server requires that the file name already exist, but the file does not yet exist on the server, a message such as the following is displayed on the PowerHub terminal:

```
18:PowerHub:tftp# savecfg ace.cfg
tftpWrite: Peer generated error
tftp: File not found: File not found
19:PowerHub:tftp#
```

CHAPTER 13 Bridge Commands

The PowerHub software contains implementations of IEEE 802.1d bridging and the 802.1d Spanning-Tree protocol. This chapter describes the bridge subsystem commands you can use to perform the following tasks:

- Show the bridge configuration
- Show and manage the bridge table (includes changing the aging interval for dynamic (learned) entries))
- Show, add, and delete bridge groups
- Show the bridging status of a PowerHub segment
- Enabling, disabling, and configuring Spanning-Tree
- Display or clear packet, bridge, and segment statistics
- Display and clear the bridge cache

13.1 Accessing the Bridge Subsystem

To access the `bridge` subsystem, issue the following command from any command prompt:

```
bridge
```

13.2 Showing the Bridging Configuration

Use the **`config`** command to display the bridge configuration parameters. The syntax for this command is:

```
config [show] [<argument-list>]
```

<argument-list> Specifies the configuration parameters you want to display. You can specify an argument, a comma-separated list of arguments, or **`all`** for all arguments. Table 13.1 lists the arguments you can specify. The default is **`all`**.

Table 13.1 - `config` command arguments

command arguments	**command descriptions**
vars	The aging time for entries in the bridge table. This also shows if learning is enabled.
groups	The currently defined network groups.
templates	All logical filtering templates that are defined.
rules	All logical filtering rules that are defined.
filters	The packet-forwarding restrictions for all segments. This includes the source and destination logical filtering rules and whether or not learned entries are blocked.
spantree|st	All parameters that are configured for the Spanning-Tree Algorithm.

NOTE

Command syntax for the **lrule** and **template** commands, are located in the *PowerHub Filters Manual*.

The following example shows the type of information displayed by the **show config** command when issued without arguments.

```
46:PowerHub:bridge# show config
Spanning Tree
  Status:            Enabled
  System Priority:   8000
  Spanning Tree Add: 01-80-c2-00-00-00
  My Bridge Address: 00-00-ef-02-42-50
  Max Age:           Curr val: 21  (Config val: 21)
  Hello Time:        Curr val: 4   (Config val: 4)
  Forward Delay:     Curr val: 16  (Config val: 16)
  Send Fast Hellos:  Disabled
  Fast Hello Parms:   HelloTime:1sec, HighUtil:70%, LowUtil:50%
  Seg Prio PathCost DesignatedBridge DesSeg DesCost StaChngs
  --- ---- -------- -----------      ------ ------- --------
  1.1 128  100      this-bridge 1      10      1
  1.2 128  100      this-bridge 2      10      1
  1.3 128  100      this-bridge 3      10      1
  1.4 128  100      this-bridge 4      10      1
  1.5 128  100      this-bridge 5      10      1
  1.6 128  100      this-bridge 6      10      1
  2.1 128  100      this-bridge 7      10      1
  2.2 128  100      this-bridge 8      10      1
  2.3 128  100      this-bridge 9      10      1
  2.4 128  100      this-bridge 9      10      1
Bridge learning
     segment 1.1: on
     segment 1.2: on
     segment 1.3: on
     segment 1.4: on
     segment 1.5: on
     segment 1.6: on
     segment 2.1: on
     segment 2.2: on
     segment 2.3: on
     segment 2.4: on
Bridge table aging time: 60 minutes
IP bridging: disabled
Bridge Groups
Name               Segment List
----               ------------
default            1.1, 1.2, 1.3, 1.4, 1.5
                   1.6, 2.1, 2.2, 2.3, 2.4
Filter templates
        Number  Offset(dec)  Mask(hex) Comparator(hex)
        099     004          00000000   00000000
Filter rules
        Number  Description
        163     99
Filters applied
     Segment     Transmit     Receive
       2.1       -            -
       2.2       -            -
       2.3       -            -
       2.4       -            -
       2.5       -            -
```

13.3 Using the Bridge Table

The *bridge table* contains information about devices attached to the PowerHub switch. The PowerHub software uses the entries in the bridge table to bridge packets. Entries are added to the table automatically or manually.

- Entries Added Automatically

Each time the PowerHub bridging engine receives a packet, it checks the packet's source address against the MAC addresses listed in the bridge table. If the address is not listed in the table, the switch adds an entry to the table. The entry contains the source device's MAC address, the segment number on which the switch received the packet, and other information used for bridging.

- Entries Added Manually

You can create a static entry using the **bt add** command. A static entry is manually added to the bridge table, rather than learned by the bridge table. Static entries are not subject to aging and remain in the bridge table until you remove them. Moreover, they are saved in the configuration file when you save the file.[1] Use static entries when you want to ensure that the PowerHub system always reaches a specific node on the same segment, or to configure a PowerHub connection to a multi-homed host.

13.3.1 Displaying the Bridge Table

To display the bridge table, issue this command:

```
bt [show] [-h] [-m] [-t] [<seglist>] [<ethaddr|ethpat>]
```

<seglist> Specifies the segment(s) for which you want to display bridge table entries. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

If you specify **all**, the entire bridge table is displayed. The default is **all**.

<ethaddr>|ethpat Is the MAC-layer hardware address of the device for which you want to display the PowerHub switch's bridge-table entries. Specify the address as six hyphen-separated two-digit hexadecimal octets (ex: **08-00-20-0f-a5-ab**).

-t Displays the total number of entries in the table. The total is comprised of the total number of learned entries and permanent (static) entries. This argument also shows how many entries remain available in the bridge pool; that is, the number of entries for which the table still has room.

-h Displays the hash displacements for the specified entries.

-m Displays entries for multi-homed hosts.

[1] To save a PowerHub configuration, issue the **system savecfg** <file-name> command.

NOTE

Because the **-h** and **-m** options display specific entries in the bridge table, they cannot be used with the **-t** option which displays total bridge entries.

Here is an example of the bridge table:

```
40:PowerHub:bridge# bt show

Bridging table (aging time = 60 minutes)
     Ethernet-address   Seg   Rule   Flags
     00-20-af-06-ee-ee  2.16  none
     01-80-c2-00-00-00   --   none  spanning-tree permanent bmcast
     00-00-ef-02-4f-d0  2.16  none
     01-00-5e-00-00-05   --   none  permanent bmcast
     01-00-5e-00-00-06   --   none  permanent bmcast
     00-a0-24-6e-3a-73  2.16  none
     08-00-87-0b-90-e4  2.16  none
     ff-ff-ff-ff-ff-ff   --   none  permanent bmcast
Total entries: 8, Learned entries: 0, Permanent Entries: 4
```

The bridge table contains the following information for each entry:

Ethernet-address The MAC-layer hardware address of the device.

Seg (Segment) The number of the segment to which the network joining the device to the hub is attached. If the MAC-layer hardware address belongs to a multi-homed host, the segment number is shown as MH.

Rule The number of a logical filtering rule applied to packets that are forwarded to or from this address. See PowerHub Filters Manual for information about defining rules.

Flags The software maintains certain flags in order to use and manage addresses in the bridge table. For example, switch entries such as the PowerHub's own address are marked, and entries that haven't been used recently are flagged for possible deletion (aging).

Each entry in the bridge table can have one or more of the following flags:

bmcastA broadcast/multicast address.

permanentMost often, this flag indicates that the address is a static entry (created using the **bt add** command). Otherwise, it is a switch-defined entry.

spanning-treeThe industry-standard (IEEE 802.1d) multicast address used by the Spanning-Tree algorithm.

systemThe factory-configured MAC-layer hardware address of the hub.

blankIn a typical application, most entries in the bridge table have none of the preceding flags set. Such entries are learned addresses that have been seen at least once since the last time the bridge table was aged.

13.3.2 Clearing the Bridge Table

Periodically, learned entries are automatically removed from the bridge table through aging. However, you can clear all learned entries from the table using this command:

```
bt clear
```

13.3.3 Adding an Entry to the Bridge Table

Use the **bt add** command to add a static (permanent) entry to the bridge table. The entry is added to the table as soon as you issue the command and remains in the table until you remove the entry. This command is helpful because adding static bridge entries is an effective way to ensure that a hub can always recognize a specific node that is permanently located on a segment. Unlike learned entries, static entries are not subject to aging. Here is the syntax for this command:

bt add [*<ethaddr>*] [*<seglist>*]

<ethaddr> Specifies the MAC address of the device.

<seglist> Specifies the segment(s) associated with the specified MAC address. To ensure that packets destined for the device are forwarded successfully, make sure you specify the segments to which the device is attached.

NOTE

If you specify more than one segment, each segment is considered to be attached to a multi-homed host, and the flag M appears in the Segment column in place of a segment number.

Here is an example of how to create a bridge table entry for a multi-homed host. In this example, the entry is defined, then the bridge table containing the newly created entry is displayed. The entry is highlighted in bold type.

```
40:PowerHub:bridge# bt add 00-00-EF-02-00-F0 1.1, 1.2, 1.3
address 08-00-20-00-ef-2b: added on 1.1, 1.2, 1.3

Bridging table (aging time = 60 minutes)
     Ethernet-address   Seg   Rule   Flags
     00-00-ef-01-93-40  10    none   permanent bmcast
     01-80-c2-00-00-00  --    none   system permanent
     aa-00-04-00-16-08  11    none   permanent bmcast
     07-00-2f-e4-b3-ee  --    none   permanent bmcast
     ab-00-00-03-00-00  --    none   permanent bmcast
     00-00-ef-01-10-20  10    none   system permanent
     aa-00-04-00-f1-33  10    none   system permanent
     ff-ff-ff-ff-ff-ff  --    none   permanent bmcast

Total entries: 9, Learned entries: 0, Permanent Entries: 9
```

13.3.4 Enabling and Disabling Bridge Learning

When bridge learning is enabled, MAC addresses from packets received on the PowerHub switch are recorded. The PowerHub switch uses the learned MAC addresses to return packets to those destinations. By default, bridge learning is enabled when you boot up the PowerHub system. To enable bridge learning, issue the following command:

learning|learn penable *<seglist>*

penable Enables bridge learning on the specified segments.

<seglist> Specifies the segment(s) that you want bridge learning enabled. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

Here are the results produced by this command:

```
40:PowerHub:bridge# learning penable 1.4
Learning enabled on segment 1.4
```

To disable bridge learning on the specified segment(s), issue the following command:

learning|learn pdisable *<seglist>*

13.3.5 Changing the Aging Interval

Use the **aging set** command to set the bridge table aging time. Aging is a mechanism that periodically clears learned entries from the table. Only dynamic entries (entries learned by the software and not configured manually by the user) are aged by the software. Static entries (those created by the user) do not age.

At an interval you specify (the aging interval), the PowerHub software determines which of the learned entries in the table have not been recently used. Each learned entry that has not been used during the specified interval is marked aged. This value shows up in the Flags column of the bridge table.

If an entry marked aged is used during the next aging interval, the aged flag is removed and the entry remains in the table. However, if an entry marked aged is unused during the next interval, the entry is removed from the table. Here is the syntax for this command:

```
aging set [<time>]
```

<time> Specifies the aging time to clear learned entries in seconds. Default is set to 60 minutes.

To unset aging enter the following command:

```
aging unset
```

13.3.6 Deleting an Entry from the Bridge Table

The **bt del** command can also be used to delete a permanent bridge entry from the bridge table. The entry is deleted by issuing this command along with the entry's Ethernet address. Here is the syntax of this command:

```
bt del [<ethaddr>]
```

<ethaddr> Specifies the Ethernet address of the entry to be deleted.

13.4 Defining and Adding Bridge Groups

Use the **pset group** command to define a network group. A network group is a specific subset of network segments among which packets can be bridged, creating a Layer-2-only VLAN. A packet from one segment in the network group can be bridged only to the other segments in the network group.

You can define up to 32 network groups. Group membership can overlap, and each segment can belong to all, some, or none of the network groups.

As shipped from the factory, the PowerHub bridging engine contains one network group known as **default**. All segments attached to the hub automatically belong to the **default** network group. The **default** group is added to your configuration file when you save the PowerHub configuration.

If you want to restrict bridging within your network, you can delete the **default** network group and define your own network groups.

NOTE

If you save the PowerHub configuration using the **system savecfg** command, the default group is automatically added to the configuration file. If your configuration requires that not all segments belong to a common network group (for example, if you defined groups with restricted sets of segments), make sure you delete the default group before saving the configuration file. To delete the default group, issue the following command: **punset group <***groupname***>**.

Here is the syntax of the **pset group** command:

```
pset group <groupname> <seglist>
```

<groupname> Specifies the name of the network group. You can specify any alphanumeric string up to 15 characters in length.

<seglist> Specifies the segment(s) that belongs to the network group. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

If you specify **all**, all segments are added to the network group.

NOTE

To create a new **default** group, you must specify **all** or list all the segments in the hub as the <seglist>. If you specify a <seglist>, instead of **all**, and the <seglist> does not include all the segments in the hub, the software creates a network group called **`old_default`**. This default group is stored in the configuration file when you save the configuration.

13.4.1 Displaying the Bridge Groups

The **`config show`** command can be used to display bridge groups configured on the PowerHub. Here is the syntax of this command:

```
config [show] groups
```

Here is an example of configured bridge groups:

```
40:PowerHub:bridge# config show groups

Bridge Groups
Name              Segment List
--------------    ----------------
default           1.1, 1.2, 1.3, 1.4, 1.5
                  1.6, 2.1, 2.2, 2.3, 2.4
```

13.4.2 Deleting a Bridge Group

Use the **`punset group`** command to delete a bridge group. Here is the syntax for this command:

```
punset group <groupname>
```

<groupname> Specifies the name of the network group you want to delete.

13.5 Displaying the Bridging Status of a Segment

Use the **`status show`** command to display the bridge status for each segment. When you issue this command, the bridge status can differ depending whether you are bridging or routing on particular segments. Here is the syntax for this command:

```
status [show]
```

Here is an example of the display produced by the **`status`** command:

```
40:PowerHub:bridge# status show
Segment      Segment Name             Spanning-tree
--------     ----------------         ------------------
1.1          Port_1                   disabled
1.2**        Port_2                   forwarding
1.3          Port_3                   disabled
1.4          Port_4                   forwarding
1.5          Port_5                   blocking
1.6          Port_6                   forwarding
2.1          Port_14                  forwarding
2.2          Port_15                  forwarding
2.3          Port_16                  disabled
2.4          Port_17                  disabled
```

For bridge or VLAN traffic to be forwarded on the segment, the Spanning-tree state must be forwarding. The Spanning-Tree state does not affect routed traffic on the segment.

Note that the Spanning-tree state blocking does not indicate a problem in your network. As described in Section 13.6.1, the Spanning-Tree algorithm breaks loops in your bridge network by blocking certain segments. The columns in this display show the following information:

Segment The segment number listed in this column corresponds to the physical location of the segment in the PowerHub chassis. Use the **`system config show`** command to display information about a segment’s physical location in the chassis. See your *PowerHub Hardware Reference Manual*, for more information about this command.

If the segment number is followed by ** (two asterisks), then bridging has been disabled by the **`bridging`** command on that segment. Note that the bridging command does not affect routing. In this example, bridging has been disabled on segments 1.1, 1.3, 2.3, and 2.4.

Segment Name The description assigned to each segment. You can change the description using the **`media sset segment name`** command. See the *PowerHub Hardware Reference Manual, V 3.0,* for more information about this command.

Spanning-tree The Spanning-Tree algorithm automatically causes segments to forward or block traffic based on the network topology. When the Spanning-Tree algorithm is enabled, this column shows one of four states:

listening
learning
blocking
forwarding
disabled

The `listening` and `learning` states occur when you first enable the Spanning-Tree feature or when your network topology changes. The `blocking` state indicates that packets are not being forwarded. The `forwarding` state indicates that packets can be forwarded on the segment. The `disabled` state indicates that the segment has been disabled using the segment command.

In this example, the Spanning-Tree feature is blocking bridge traffic on segment 1.5. The Spanning-Tree state has no effect on routing. However, this state does affect VLANs because traffic is bridged within VLANs, rather than routed.

13.6 Configuring Spanning-Tree Parameters

The Spanning-Tree algorithm is a mechanism that logically eliminates physical loops in a bridged network. For example, if your bridges are configured in such a way that broadcast/ multicast packets are eventually forwarded back to the bridge that first sent them, your network has a loop. Unless you reconfigure your network topology or your bridges to break the loop, or implement a mechanism to logically break the loop, broadcast/multicast packets are forwarded from bridge to bridge indefinitely, clogging your network. Whenever a segment's state is changed, either by automatic segment-state detection or by a user-interface command, the Spanning-Tree algorithm adjusts the network topology accordingly.

When the Spanning-Tree algorithm is enabled using the spantree command (see Section 13.6.1), you can fine-tune the following Spanning-Tree parameters:

- Bridge priority
- Segment priority
- Timer threshold
- Spanning-Tree path cost
- Fast hello-time thresholds (if the fast hello-time feature is enabled)

The first four parameters always are used; the last one is optional. The following sections describe how to adjust these parameters. To display the current settings for these parameters, issue the following command:

```
config [show] st
```

Here is an example of the display produced by the **config [show] st** command:

```
46:PowerHub:bridge# config show st
Spanning Tree
  Status:             Enabled
  System Priority:    8000
  Spanning Tree Add: 01-80-c2-00-00-00
  My Bridge Address: 00-00-ef-02-42-50
  Max Age:            Curr val: 21  (Config val: 21)
  Hello Time:         Curr val: 4   (Config val: 4)
  Forward Delay:      Curr val: 16  (Config val: 16)
  Send Fast Hellos:   Disabled
  Fast Hello Parms:    HelloTime:1sec, HighUtil:70%, LowUtil:50%

  Seg Prio PathCost DesigBridge DesSeg DesCost StaChngs
  --- ---- -------- ----------- ------ ------- --------
  1.1 128  100      this-bridge 1      10      1
  1.2 128  100      this-bridge 2      10      1
  1.3 128  100      this-bridge 3      10      1
  1.4 128  100      this-bridge 4      10      1
  1.5 128  100      this-bridge 5      10      1
  1.6 128  100      this-bridge 6      10      1
  2.1 128  100      this-bridge 7      10      1
  2.2 128  100      this-bridge 8      10      1
  2.3 128  100      this-bridge 9      10      1
  2.4 128  100      this-bridge 9      10      1
```

13.6.1 Enabling or Disabling Spanning Tree

To enable the Spanning-Tree algorithm, issue the following command:

```
spantree enable|disable
```

enable|disable Specifies whether you are enabling or disabling the Spanning-Tree algorithm. The default is **disable**.

13.6.2 Changing Spanning-Tree Parameters

This section describes how to change the setting of individual Spanning-Tree parameters.

13.6.2.1 Setting the Bridge Priority

Use the **spantree set bridge-priority** command to adjust the bridge priority. Here is the syntax for this command:

spantree|st set bridge-priority|bp *<priority>*

<priority> Is a hexadecimal number in the range from **0** through **ff**(hex). The default is **80**(hex).

The setting for the bridge priority is displayed in the System Priority field of the **config show st** display.

13.6.2.2 Setting a Segment's Priority

Use the **spantree set seg-priority** command to adjust the bridge priority for a segment or a list of segments. Here is the syntax for this command:

spantree|st sset seg-priority|sp *<priority> <seglist>*

<priority> Is a hexadecimal number in the range from **0** through **ff**(hex). The default is **80** (hex). You must specify a separate priority for each segment.

<seglist> Specifies the segments for which you are setting the priority. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

Here is an example of this command. Notice that a separate priority is specified for each segment in the segment range. Even if you plan to assign the same priority to all the segments, you must list the priority individually for each segment.

```
27:PowerHub:bridge# spantree set seg-priority 0 1.2
Ok. Spanning tree has been reinitialized
```

The segment priority for each segment is displayed in the Priority field of the **config show st** display.

13.6.2.3 Setting the Path Cost

You can adjust the Spanning-Tree cost to reconfigure the shortest path to the root bridge. Use the **spantree set path-cost** command to adjust the Spanning-Tree cost on a per-segment basis. Here is the syntax for this command:

spantree|st set path-cost|pc *<path-cost> <seglist>*

<path-cost> Specifies the cost of the path. You can specify a value from 1 through 65535. The default is 100 for 10Mb/s Ethernet segments, and 10 for FDDI and Fast Ethernet segments. You must specify a separate path cost for each segment.

<seglist> Specifies the segments for which you want to adjust the Spanning-Tree cost. You can specify a single segment or a comma-separated list of segments.

Here is an example of this command. Notice that a separate path cost is specified for each segment in the segment range. Even if you plan to assign the same path cost to all the segments, you must list the path cost individually for each segment. In the following example, the path cost 90 is assigned to segment 1. 3.

```
27:PowerHub:bridge# set path-cost 90 1.3
```

The path cost for each segment is displayed in the Path Cost field of the **`config show st`** display.

13.6.2.4 Setting the Maximum Age

Use the **`spantree set maxage`** command to adjust the maximum age of the bridge-timer threshold. Here is the syntax for this command:

```
spantree|st set maxage <time>
```

<time> Specifies the maximum age, in seconds. You can specify from **6** through **40** seconds. The default is **21** seconds.

The maximum age of the bridge timer threshold is displayed in the Max Age field of the **`config show st`** display.

13.6.2.5 Setting the Hello Time

Use the **`spantree set hello`** command to adjust the hello time of the bridge-timer threshold. Here is the syntax for this command:

```
spantree|st set hello <time>
```

<time> Specifies the hello time, in seconds. You can specify from **1** through **10** seconds. The default is **4** seconds.

To display the current setting, issue the **`config show st`** command to display the Spanning-Tree settings, then check the value in the Hello Time field.

13.6.2.6 Setting the Forward Delay

Use the **`spantree set fwddelay`** command to adjust the forward delay of the bridge-timer threshold. Here is the syntax for this command:

```
spantree|st set fwddelay <time>
```

<time> Specifies the forward delay, in seconds. You can specify from **4** through **30** seconds. The default is **16** seconds.

The forward delay of the bridge timer threshold is displayed in the Priority field of the **`config show st`** display.

13.6.2.7 Setting the Fast-Hello Time

Under heavy network traffic, Spanning-Tree hello packets are not transmitted at regular hello-time intervals. Such irregular time intervals can delay the transmission of hello packets. If hello packets are delayed past a certain time value, called the maximum age, your Spanning-Tree state can change. If the segment state is “blocking,” and hello packets are not received before another time value, the Max Age, your Spanning-Tree state will change to “listening” and then to “learning.”

Use the **`spantree set fast-hello`** command to enable or disable the fast hello timer feature. Here is the syntax for this command:

```
spantree|st set fast-hello <time>
```

This feature is disabled by default. To display the current setting, issue the **`config show st`** command to display the Spanning-Tree settings, then check the value in the Sending Fast Hellos field.

13.6.2.8 Setting the High- and Low-Utilization Percentage

If the fast hello timer feature is enabled, when a segment’s utilization exceeds an upper-end value (<*high-util*>), the PowerHub software automatically compensates for the increased traffic by using fast hello time to transmit hello packets. The fast hello time is less than the normal (configured) hello time. When all segments’ utilizations drop below a lower-end value, the <*low-util*>, the hello time reverts to normal (either previously configured or system defaults).

Use the **`spantree set high-util`** command to set the high-utilization threshold. Here is an example of this command:

```
spantree|st set high-util <percentage>
```

<percentage> Specifies the upper-end value of segment utilization. If segment utilization exceeds this value and the fast hello timer feature is enabled, PowerHub software automatically compensates for the increased network traffic. This value is a percentage in the range of **1** to **100**. The default is **70**%.

Use the **spantree set low-util** command to set the low-utilization threshold. Here is an example of this command:

```
spantree|st set low-util <percentage>
```

<percentage> Specifies the lower-end value of segment utilization. If segment utilization drops below this value, the PowerHub software automatically reverts to normal hello time. This value is a percentage in the range of **1** to **100**. The default is **50%**.

13.7 Displaying and Clearing Bridge Statistics

Use the **stats show** command to display table misses, a subset of statistics, or all statistics. Here is the syntax for this command:

```
stats [show]
```

13.7.1 Clearing Statistics

Use the **stats clear** command to clear all bridge statistics. This command resets all bridge statistics to 0, then begins collecting statistics again. Once you clear the bridge statistics, the statistics displayed in response to the **stats** command show the counts since the most recent clear, rather than since the most recent reboot.

13.8 Displaying and Clearing the Bridge Cache

The PowerHub software maintains a bridge cache. Each time the bridging engine bridges a packet, it creates an entry in the bridge cache containing the packet's destination Ethernet address and source Ethernet address. The bridge cache is frequently updated with the most-recently used source-destination pairs and provides a fast path for bridge traffic resulting in increased performance. You can use the bridge cache for at-a-glance information about the current bridge traffic in your network.

13.8.1 Displaying the Bridge Cache

To display the bridge cache, issue the following command:

```
cache [show] <seglist>
```

<seglist> Specifies segments for which you want to display the cache entries. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

Here is an example of the bridge cache:

```
17:PowerHub:bridge# display-cache
Bridging cache:
Port 01: Dest: 08-00-20-08-70-54, Source: 08-00-20-0f-dd-99
         Dest: 00-00-6b-82-3f-34, Source: 08-00-20-0f-6c-96
         Dest: 08-00-20-08-85-69, Source: 08-00-20-0f-dd-99
         Dest: 08-00-20-08-70-54, Source: 08-00-20-0f-6c-96
Port 02: Dest: 00-00-6b-82-3f-34, Source: 08-00-20-0e-ae-03
         Dest: 00-00-94-06-79-12, Source: 08-00-20-10-56-53
         Dest: 08-00-20-0f-f2-9d, Source: 08-00-20-0e-ae-03
         Dest: 00-00-c0-ed-61-4a, Source: 08-00-20-10-56-53
         Dest: 08-00-20-08-85-69, Source: 00-00-6b-82-3f-34
         Dest: 08-00-20-08-70-54, Source: 08-00-20-0e-ae-03
Listing continues
Port 21: empty
Port 22: empty
Port 23: empty
Port 24: empty
```

13.8.2 Clearing the Bridge Cache

To ensure that the entries displayed in the bridge cache are recent and reflect current traffic patterns, you can clear the cache just before displaying it. To clear the bridge cache, issue the following command:

```
cache clear
```

The entire contents of the bridge cache are removed.

13.9 IPX Translation Bridging

IPX translation bridging lets you configure one or more IPX networks that span across FDDI and Ethernet segments using different packet encapsulations. Without altering the configurations of individual devices, IPX translation bridging enables Ethernet and FDDI devices with different encapsulation types to communicate with each other. This feature is especially useful if your IPX network consists largely of Ethernet devices using 802.3 encapsulation, the default encapsulation type in Novell IPX software versions 2.2 through 3.11.[1] However, if your network name is not in the IBT table, IPX translation bridging does not occur, and normal bridging does.

This section describes how to enable IPX translation bridging as well as how to add, display, or delete IPX translation-bridging networks.

NOTE

IPX translation bridging is independent of IPX routing—they are mutually exclusive. We recommend that you do not enable both IPX translation bridging and IPX routing. However, if both IPX translation bridging and IPX routing are enabled, IPX routing takes precedence over IPX translation bridging.

13.9.1 Encapsulation Types

When you use IPX translation bridging, you specify the Ethernet and FDDI encapsulation types to be used on each IPX network. For each IPX network number, you can specify both the Ethernet and FDDI encapsulations you want the software to use on that network. Table lists the combinations of encapsulation types you can specify.

[1] If your FDDI device does not support 802.3, you cannot bridge between the Ethernet devices and the FDDI device using standard IPX bridging. You must use IPX Translation bridging in this case.

Table 13.2 - Valid Encapsulations for IPX Translation Bridging

	ENET	802.2	802.3*	SNAP
FDDI		4	4	4
Ethernet	4	4	4	4

* The FDDI “raw” encapsulation is 802.3-like and is listed as “802.3” in table and command descriptions. However, this encapsulation is not identical to the 802.3 format on Ethernet since it does not include an explicit length field. See Appendix C for the format of each type of encapsulation.

For further information about IPX bridging over FDDI and packet encapsulation information, see Appendix C.

13.9.2 Configuration Requirements

Although IPX translation bridging is simple to configure, the following conditions must be met:

- The servers attached to the segments in an IPX translation bridging network must be configured to have the same network number as the “IPX translation-bridging” network number configured on the hub. If a server’s network number cannot be changed to correspond to the IPX translation-bridging network you define on the hub, you can change the network number defined on the hub to match the server.
- Servers and clients must be configured to have the same encapsulation type as the type specified for the appropriate medium in your IPX translation-bridging network. For example, a client attached to an Ethernet segment must be configured to use the same Ethernet encapsulation type as the one defined for the corresponding IPX translation-bridging network. However, if encapsulation types on the server or client cannot be changed, the encapsulation types of the client or server can be configured on the hub.

13.9.3 Enabling IPX Translation Bridging

Before you can use the IPX translation-bridging feature, you must enable IPX translation bridging. To enable IPX translation bridging, issue the following command:

```
ipx-br-translation|ibt enable|disable
```

enable|disable Specifies whether you are enabling or disabling IPX translation bridging.

Here is an example of the use of this command:

```
40:PowerHub:bridge# ipx-br-translation enable
IPX translation bridging is now enabled
```

13.9.4 Adding an IPX Translation-Bridging Network

To create an IPX translation-bridging network, use the following command:

ipx-br-translation|ibt add *<network> <ethernet-encap> <fddi-encap>*

<network> Specifies the IPX network number to which you are applying the encapsulation settings.

<ethernet-encap> Specifies the encapsulation type to be used for Ethernet packets. Packets bridged from FDDI to this network number are converted to this encapsulation.

<network> Specifies the encapsulation type to be used for packets translated to FDDI.

NOTE For further information about valid IPX bridging encapsulation formats, see Appendix C.

Here are some examples of how to use this command. In these examples, definitions are created for IPX translation-bridging networks 100, 200, and 300:

```
2:PowerHub:bridge# ipx-br-translation add 100 802.2 snap
IPX network 100 added to the translation table
3:PowerHub:bridge#  ipx-br-translation add 200 802.2 802.2
IPX network 200 added to the translation table
4:PowerHub:bridge#  ipx-br-translation add 300 802.3 snap
IPX network 300 added to the translation table
```

13.9.5 Displaying IPX Translation-Bridging Information

At any time, you can display the definitions for the IPX translation-bridging networks defined on the hub. To display the definitions, use the following command:

ipx-br-translation|ibt [show] [*<network>*]|[-t]

<network> Specifies an IPX translation-bridging network number.

<-t> Displays only the total number of entries in the IPX translation-bridge table.

Here are some examples of displays produced by this command. In the first example, no specific network number is given, so all individual entries are displayed, as well as the total number of entries. In the second example (prompt 7), the **-t** argument is used to display the total number of IPX translation-bridging entries.

```
6:PowerHub:bridge# ipx-br-translation show
IPX Translation Bridging: Enabled
IPX Network        Ethernet Encap        FDDI Encap
-----------        --------------        ----------
100        802.2                 802.2/SNAP
200        802.2                 802.2
300        Ethernet II           802.2/SNAP
Total entries:  3
7:PowerHub:bridge# ipx-br-translation show -t
IPX Translation Bridging: Enabled
IPX Network        Ethernet Encap        FDDI Encap
-----------        --------------        ----------
Total entries:  3
```

13.9.6 Deleting IPX Translation-Bridging Information

To delete the encapsulation settings assigned to a network number, use the following command:

ipx-br-translation|ibt delete *<network>*|all

<network>|all Specifies an IPX translation-bridging network number. If you specify **all**, all IPX translation-bridging networks are deleted.

Here is an example of the use of this command:

```
8:PowerHub:bridge# ipx-br-translation delete all
All IPX networks deleted from the IPX translation table
```

CHAPTER 14 SNMP Commands

The PowerHub contains an implementation of Simple Network Management Protocol (SNMP). SNMP uses User Datagram Protocol (UDP), an industry-standard connectionless protocol used to send and receive packets between a managed hub and other devices. This chapter describes the commands in the `snmp` subsystem and shows how to perform the following tasks:

- Display the SNMP configuration.
- Add an SNMP management community.
- Add an SNMP manager.
- Delete an SNMP management community.
- Delete an SNMP manager.
- Display SNMP packet statistics.
- Clear SNMP packet statistics.

In addition, this chapter describes how to set up files for use with SunNet Manager to access the PowerHub Management Information Bases (MIBs).

Using a third-party SNMP application, the PowerHub MIB objects can be accessed for information about the PowerHub. The software contains implementation of standard MIBs and the PowerHub Proprietary MIB.

14.1 Accessing the SNMP Subsystem

To access the `snmp` subsystem, issue the following command at the runtime command prompt:

```
snmp
```

14.2 Displaying the SNMP Configuration

To display the current configuration of communities and managers, use the **`config show`** command.

`config [show] [-l] [`*<community-name>***`]`**

-l Specifies that managers and trap configurations are to be listed.

<community-name> Specifies the community for which you want configuration information displayed.

When used with no arguments, this command lists only communities and each community's access (the first and second column from the example shown below). The example that follows illustrates the **`config show`** command containing both arguments. It shows the manager and trap configuration for a specific community.

```
47:PowerHub:snmp# config show -l admin

Community            Access     Managers          Traps
-------------------  ------     -------------     ------
admin                  rw       147.128.7.3       notrap
```

14.3 Displaying Statistics

The SNMP subsystem maintains statistics on SNMP packets that are transmited and received. These statistics are a superset of the corresponding statistics provided in the SNMP table of MIB-II. The PowerHub maintains two copies of each SNMP statistics counter:

- Count since last clear.
- Count since last switch reset.

To display these statistics, use the following command:

`stats [show] [-t]`

-t Displays statistics since the last reset.

The following example shows the **stats** command used without the **[-t]** argument:

```
10:PowerHub:snmp# stats
SNMP packet statistics (count since last stats clear):
Packets Rcvd:           49086    Packets Sent:           49086
Bad Version Rcvd:           0    Bad Comm Name Rcvd:         0
Bad Comm Uses Rcvd:         0    ASN Parse Err Rcvd:         0
Bad Type Rcvd:              0    Too Big Rcvd:               0
No Such Name Rcvd:          0    Bad Values Rcvd:            0
Read Onlys Rcvd:            0    Gen Errs Rcvd:              0
Total vars Req:         49086    Total vars Set:             0
Get Req Rcvd:               0    GetNext Req Rcvd:       49086
Set Req Rcvd:               0    Get Resp Rcvd:              0
Traps Rcvd:                 0    Too Big Sent:               0
No Such Name Sent           0    Bad Values Sent:            0
Read Onlys Sent:            0    Gen Errs Sent:              0
Get Req Sent:               0    GetNext Req Sent:           0
Set Req Sent:               0    Get Resp Sent:          49806
Traps Sent:                0
```

14.3.1 Clearing Statistics

To clear statistics, use the **stats clear** command as in the following example:

```
51:PowerHub:snmp# stats clear
Okay
```

14.4 Adding an SNMP Community

The default configuration includes the standard default SNMP community, **public**, which has read-only access. Other communities can be added with the **community add** command. The syntax for the **community add** command is:

community|com add *<community-name>* **[ro|rw]**

<community-name> Specifies the community to add.

[ro|rw] Specifies the community's access as read-only (**ro**) or read-write (**rw**). The default is read-only access.

The following example illustrates adding an **admin** community with read-write access.

```
43:PowerHub:snmp# community add admin rw
```

A maximum of eight SNMP communities are supported.

14.4.1 Deleting an SNMP Community

Communities are deleted using the **community delete** command. The syntax for this command is:

community|com delete|del <*community-name***>**

<community-name> Specifies the community to delete.

This command deletes both the community name and all managers associated with it. The following example illustrates deleting the **admin** community.

```
44:PowerHub:snmp# community delete admin
```

14.4.2 Supported SNMP Traps

The PowerHub supports the following standard traps, as defined in RFC 1157:

- coldStart
- warmStart
- linkDown
- linkUp
- authenticationFailure

14.4.3 Adding an SNMP Manager

Each community can include up to 16 managers. Managers are added with the **manager add** command. The syntax for the **manager add** command is:

manager|man add <*community-name***> <***IP-addr***> [trap|notrap]**

<community-name> Specifies the community name for which to add a manager.

<IP-addr> Specifies the IP address of the manager.

trap|notrap Is an optional flag, indicating whether the manager should receive traps. If the manager should receive traps, use (**trap**). If the managershould not receivetraps, use (**notrap**). The default is **notrap**.

The following example illustrates how to delete both the community name and all managers associated with it. The command that follows illustrates deleting the **admin** community.

```
44:PowerHub:snmp# community delete admin
```

In the following example, a manager with IP address 147.128.7.3 is added to the **`admin`** community.

```
44:PowerHub:snmp# manager add admin 147.128.7.3 notrap
```

14.4.4 Deleting an SNMP Manager

To delete a manager, use the **`delete manager`** command. The syntax for the **`delete manager`** command is:

```
manager|man delete|del <community-name> <IP-addr>|all
```

<community-name> The community name of the manager to delete.

<IP-addr>|all The IP address of the manager to delete. If **`all`** is specified, all managers in the community are deleted, but the community itself remains configured.

14.4.5 Preparing Files for SunNet Manager

If SunNet Manager is being used to access the PowerHub MIBs, the following types of files must be prepared for each MIB:

- Schema
- Trap
- OID.

Table 14.1 lists the utilities and file names in SunNet Manager used to prepare these files.

Table 14.1 - SunNet Manager Utilities.

schema	A MIB converted from ASN.1 format.	mib2schema	*`<MIB-name>`*.schema
trap	Active traps for a particular MIB.	mib2schema	*`<MIB-name>`*.trap
OID	Object Identify file. Translates the Object Identifiers used by SNMP to communicate into the identifiers that SunNet Manager understands.	mib2schema	*`<MIB-name>`*.oid
*Where *`<MIB-name>`* is the name of the MIB.			

CHAPTER 15 Configuring IP Routing

This chapter describes the commands in the `ip` subsystem and shows how to use them to configure and manage the PowerHub as an IP router. Using `ip` subsystem commands, the following can be accomplished:

- Display the IP configuration
- Add, show, and delete IP interfaces
- Enable IP routing (and allocate additional memory to the IP route table)
- Add, show, and delete IP routes
- Enable, show current settings for, and change the configuration of Router-Discovery.
- Show, add static entries to, and delete static entries from the IP Address Resolution Protocol (ARP) table.
- Ping IP workstations or other IP routers
- Add and delete IP helper addresses
- Customize the routing behavior
- Show and clear IP, ICMP, ARP, RIP, and IP Helper statistics
- Show or clear the IP route cache

15.1 Accessing the IP Subsystem

To access the `ip` subsystem, issue the following command at the runtime command prompt:

```
ip
```

15.2 Displaying the IP Configuration

The current IP configuration can be displayed using the **`config [show]`** command. Following is an example of the display produced by this command:

```
117:PowerHub:ip# config show
IP Configuration:
----------------
IP Forwarding:                     enabled (gateway)
Load Balancing:                    Off
Default TTL:                       64
Arp cache aging time:              5:00
Routing Network Broadcasts:        enabled
VLAN Bridging Network Broadcasts:  enabled
Routing Broadcast Packets:         enabled
Send ICMP redirects:               enabled
Forward Pkts with SrcRt Option:    enabled
Arp auto-learn:                    enabled (gateway)
Arp VLAN Strict:                    enabled (gateway)
Routed Packet Snooping:            disabled
```

Any of the IP configuration items listed in this display can be set.

IP Forwarding Indicates whether IP forwarding is enabled or disabled. (See Section 15.3.9.)

Load Balancing Enables the PowerHub to distribute IP traffic to remote destinations among up to four equal-cost routes.

When load balancing is enabled, up to four load-balancing slots are used per destination to identify next-hop gateways. Packets are hashed to a slot according to source and destination IP address so that packets belonging to a given flow always take the same path.

Default TTL Indicates the time-to-live (TTL) parameter. This parameter specifies how long a packet is allowed to remain in the net before it is dropped. (See Section 15.8.3.)

ARP cache aging time Indicates when unused learned entries in the ARP table are removed from the ARP table if they continue to be inactive. (See Section 15.6.2.)

Routing Network Broadcasts Indicates whether routing of network broadcast packets in a subnetted environment is enabled or disabled. (See Section 15.8.6.2.)

VLAN Bridging Network Broadcasts Indicates whether bridging of network broadcast packets over a VLAN is enabled or disabled.

Routing Broadcast Packets Indicates whether routing of network broadcast packets addressed to the PowerHub Ethernet address is enabled or disabled. The default is "enabled."

Send ICMP redirects Indicates whether Internet Control Message Protocol (ICMP) redirect messages are enabled or disabled.

Forward Pkts with SrcRt Option Indicates whether the software is permitted to forward IP packets containing source route options.

ARP auto-learn Indicates whether the software is automatically learning ARP entries. (See Section 15.6.)

Routed Packet Snooping Indicates whether packet snooping is enabled or disabled. Packet snooping allows the IP filtering cache to be examined for information about routed IP packets.

15.3 Configuring and Showing IP Interfaces

Before the PowerHub can be used as an IP router, an IP address must be assigned to each segment through which IP packets are to be routed. When discussing TCP/IP, a connection to a physical segment is called an *interface*.

Multiple IP addresses can be assigned to the same segment. In addition, a Virtual Local Area Network (VLAN) can be created by assigning the same IP address to multiple segments. By default, IP packets are routed among different subnets, but bridges IP packets among segments on the same subnet.

When an IP interface is configured (using the **`add interface`** command), the PowerHub automatically sets the MTU value for the IP interfaces based on the medium type:

- FDDI segments, the MTU is set to 4050.
- Ethernet segments, the MTU is set to 1500.
- If the interface spans multiple segments, and those segments include both Ethernet and FDDI, the MTU value is set to 1500.

Before configuring an IP interface, read the considerations and restrictions in Section 15.3.1 and Section 15.3.2. For information about adding IP interfaces, see Section 15.4.

NOTE

If the PowerHub is configured to listen to RIP broadcasts on a subnetwork, but an IP interface address is not added to do so, a directly-attached subnet can be added.

15.3.1 Considerations

The following considerations apply to assigning interface addresses.

- An interface address must be specified in dotted-decimal notation, and it must be a valid IP host address. A valid IP address must contain a host number that is non-zero and non-broadcast (broadcast IDs are all binary 1s).
- When an IP interface is added, a subnet mask containing all ones or all zeroes can be specified.
- When an interface address is assigned to a segment, the routing software assumes that the segment is physically connected to a net whose IP network number equals the <*network-number*> part of the interface address. Routing occurs between networks with different network numbers.

NOTE

Unlike other devices, the PowerHub allows the same IP network number to be assigned to multiple segments (creating a VLAN). When this is done, IP packets are bridged among like-numbered nets that are connected to physically distinct segments.

- The PowerHub allows multiple interface addresses with different network numbers to be assigned to a single segment. When this is done, the software forwards packets for any of the corresponding nets to that segment.
- Even if routing is not desired, an interface address must be assigned to a segment in order for TELNET or SNMP connections to be made through that segment. A remote workstation uses this interface address when establishing a TELNET or SNMP connection to the PowerHub.

15.3.2 Restrictions

The following restrictions apply when IP interface addresses are assigned. These restrictions are necessary to ensure reliable operation. Invalid configurations can bring down an entire network.

- When a single network number appears on multiple segments, all those segments must be assigned the same interface address and subnet mask.
- A parent network address cannot be configured when one or more subnets of that address have been configured on one or more segments.
- A subnet address cannot be configured if its parent network address has been configured on one or more segments. The parent network is the overall network on which subnetworks are configured. For example, network 147.128.0.0 is the parent network of subnetworks 147.128.1.0 and 147.128.2.0. These two subnetworks are referred to as children networks of the parent network.
- Different IP host addresses cannot be assigned to one interface on the same network or subnet.
- For proper operation under RIP, subnet addresses should normally all have the same binary length—in other words, they should all use the same subnet mask. If it is necessary to assign variable-length subnet addresses (different subnet masks for some addresses), certain rules must be observed. For information on these rules, see *PowerHub Filters Reference Manual.*

15.3.3 How IP Packets are Handled

- Unexpected IP broadcast packets are discarded. The IP broadcast software traps IP broadcast packets and discards them immediately if they were not expected. This feature is particularly beneficial for large networks that experience high volumes of broadcast traffic.
- IP broadcast packets that are bridged back to it are discarded. Some workstations bridge broadcast packets (including RIP packets) sent from the PowerHub back to the PowerHub. The PowerHub IP software checks the IP source address of the incoming packet to determine whether the packet came from the PowerHub itself. If the packet did come from the PowerHub, the packet is discarded.
- The software routes IP broadcast packets addressed to the PowerHub Ethernet address. If routing of IP broadcast packets addressed to the PowerHub Ethernet address need to be disabled, use the **`enable|disable route-net-broadcast`** command. (See Section 15.8.6.)

15.3.4 Showing the IP Interface Table

The **`interface [show]`** command to is used display the configured IP interface addresses. For each segment, the table lists the IP addresses assigned to the segment, the link state of the segment (UP or DOWN), and other information. The syntax for this command is:

`interface|it [show] [-s] [`*<disprestrictors>***`]`**

-s Displays additional statistics, including the number of packets and octets transmitted to and received from the net by each interface.

<disprestrictors> Specifies segments for which to display the IP addresses. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

Here are some examples of the use of this command.

```
25:PowerHub:ip# interface show
        Vlan Seg   InterfaceAddr   SubnetMask     bcast  MTU   state  cost
                       3    1.3   147.128.132.1   255.255.252.0  br0    1500  down   0
        3    1.4   147.128.132.1   255.255.252.0  br0    1500  up     0
        3    1.5   147.128.132.1   255.255.252.0  br1    1500  down   5
```

By using the **`-s`** argument with this command, certain routing statistics can also be viewed, including the number of packets and octets transmitted to and received from the net by each interface.

```
26:PowerHub:ip# interface show -s
Vlan Seg   InterfaceAddr   SubnetMask     bcast  MTU   state  cost
3    1.3   147.128.132.1   255.255.252.0  br0    1500  down   0
  0 pkts in, 0 octets in, 0 pkts out, 0 octets out
3    1.4   147.128.132.1   255.255.252.0  br0    1500  up     0
  0 pkts in, 0 octets in, 0 pkts out, 0 octets out
3    1.5   147.128.132.1   255.255.252.0  br1    1500  down   5
  0 pkts in, 0 octets in, 0 pkts out, 0 octets out
```

15.3.5 Adding an IP Interface

The **interface add** command to is used assign an IP address to a PowerHub segment. When an interface address is added, the software makes an entry into the IP route table to show that the corresponding network is connected to the specified segment. The software then creates the interface. The syntax for this command is:

```
interface|it add <vlanid> <ipaddr> [/<prefixlen>|<mask>] [br[oadcast]
                        0|1] [met[ric] <metric>]
```

<vlanid> Specifies the VLAN ID to assign to the specified segment(s). By assigning the same IP address to multiple segments, a VLAN can be created. The IP address must be in dotted-decimal notation (four decimal numbers in the range 0–255 separated by dots).

<ipaddr> Specifies the IP address to assign to the specified segment(s). The IP address must be in dotted-decimal notation (four decimal numbers in the range 0–255 separated by dots).

<prefixlen> Allows a valid variable-length subnet to be created by using the **interface add** command. For more information about variable-length subnets, see *Chapter 17, Configuring IP/RIP*.

<mask> Allows a standard IP subnet mask to be used. If a particular network uses IP subnet addressing, then the subnet mask should be specified here using dotted-decimal notation. Otherwise, a default subnet mask equal to the "natural" subnet mask for the particular class of address is used.

[br[oadcast]0|1] Specifies the style of broadcast address on a segment-by-segment basis:

When **br0** is specified, an "all-0s" broadcast is sent. This means all bits in the host segment of the address are 0s. The **br0** argument is useful when the PowerHub interoperates with workstations that use the old style of IP broadcast address, with all-0s as the host number.

When **br1** is specified, the a standard "all-1s" broadcast is sent. This means all bits in the host segment of the address are 1s. The default is **br1**.

[met[ric] <metric> Specifies an additional cost of using the subnet interface. This cost is the number of extra hops to the destination. The range is 1 through 14. (The router decrements an IP packet's time-to-live field at each hop.) The default is zero. When the PowerHub reports this subnet using RIP, it adds the additional cost to the reported metric.

The cost parameter can provide controlled routing in the presence of redundant paths, such as when two PowerHubs are connected in parallel for redundancy. The cost of the attached subnets can be set to a value greater than zero in one of the PowerHubs. When the cost is set to a value greater than zero, routing is forced through the other PowerHub, if it is alive. See *Chapter 17, Configuring IP/RIP* for a description of RIP and routing operations with redundant paths.

The following example below shows the use of the **`interface add`** command to add an interface:

```
20:PowerHub:ip# interface add 1 192.12.20.17
Adding net 192.12.20.0: Okay
Port 1, Addr 192.12.20.17, Mask 255.255.255.0, added
```

Before adding the interface address, an entry is made in the route table to show that the corresponding network (192.12.20.0) is directly connected to the specified segment.

The following example shows how to add a single interface address to multiple segments using one command:

```
22:PowerHub:ip# it add 2.2, 2.4 147.128.132.1 255.255.252.0
Adding subnet 147.128.132.0: Okay
Port 2.2, Addr 147.128.132.1, Mask 255.255.252.0, added
Port 2.4, Addr 147.128.132.1, Mask 255.255.252.0, added
```

An interface address can be assigned with a non-zero cost to force routing through a desired path in the presence of redundant paths. In the following example, segments 1 and 2 are physically connected to the same router:

```
22:PowerHub:ip# it add 1.1 147.128.132.1 255.255.255.0
Adding subnet 147.128.132.0: Okay
Port 1.1, Addr 147.128.132.1, Mask 255.255.255.0 added
23:PowerHub:ip# it add 1.2 147.128.136.1 255.255.255.0 cost 3
Adding subnet 147.128.136.0: Okay
Port 1.2, Addr 147.128.136.1, Mask 255.255.255.0, cost 3, added
```

Because a higher cost is assigned to segment 1.2, all routing is forced through segment 1.1.

NOTE

When making changes to an IP address or subnet mask, it is not necessary to reboot the PowerHub.

15.3.6 Deleting an IP Interface

The **`interface del`** command is used to delete one or more interface addresses. The syntax for this command is:

`interface del[ete] [-p] <`*`vlanid`*`>|all <`*`ipaddr`*`>|all`

[-p] Allows the address-based parameters of RIP entries to be preserved.

<vlanid>|all Specifies the VLAN IDs for which to delete the corresponding interfaces. Specify a single VLAN ID or a comma-separated list of VLAN IDs. If **`all`** is specified, all VLANs are deleted from the specified segments.

<ipaddr>|all Specifies the IP addresses for which to delete the corresponding interfaces. Specify a single address, a comma-separated list of addresses, or a hyphen-separated range of addresses. If **`all`** is specified, all IP addresses are deleted from the specified segments.

NOTE

When the last interface to a particular net is deleted, that net is automatically deleted from the route table.

Following is an example of the use of this command. To delete a particular interface address on a particular segment, specify the segment number and interface address.

```
31:PowerHub:ip# interface delete 3.1 147.128.132.1
Interface address 147.128.132.1, Port 3.1: deleted
```

NOTE

When making changes to IP address or subnet mask, it is not necessary to reboot the PowerHub.

15.3.7 Configuring VLANs

To make managing network segments easier, the PowerHub allows VLANs to be created. A VLAN is a network that spans two or more physical segments. VLANs make network configuration changes simple by creating and changing LANs logically using software commands, as opposed to physically moving segment cables.

Any number of segments in the PowerHub can be defined as members of a VLAN. VLANs can overlap, so the same segments can be members of more than one VLAN. Multiple VLANs can even be defined on the same segment.

For each segment in the VLAN, the effective bandwidth available to nodes on the VLAN increases. For example, a VLAN containing six 10 Mb/s Ethernet segments enjoys 60 Mb/s of bandwidth. Even though bandwidth is increased, administration and management overhead for the segments in the VLAN does not increase, because the segments can be managed as a single network.

To add a segment to a VLAN, create an interface that contains that segment along with the other segments to be placed in the VLAN. To add a VLAN, issue the **vlan add** command. The syntax for this command is:

vlan add ***<vlanid> <seglist>***

<vlanid> Specifies the VLAN IDs t to create the corresponding VLANs. Specify a single VLAN ID or a comma-separated list of VLAN IDs.

<seglist> Specifies the segments to be included in the VLAN. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

In the following example, vlan 2 has been added to segments 1.2, 1.3, and 1.4. The **vlan show** command is used to display the results of the VLAN created:

```
31:PowerHub:ip# vlan add 2 1.2,1.3,1.4
32:PowerHub:ip# vlan show
2               1.2,1.3,1.4
```

15.3.7.1 Changing VLAN Configurations

Because VLANs are created using software commands, rather than by rearranging network segments, VLANs can easily be changed to suit networking needs. To change a VLAN, issue the **`vlan tset`** command. The syntax for this command is:

```
vlan tset <vlanid> seglist <seglist>
```

<vlanid> Specifies the VLAN IDs of the corresponding VLANs to be changed. Specify a single VLAN ID or a comma-separated list of VLAN IDs.

seglist Specifies the segments associated with the VLAN to change.

<seglist> Specifies the individual segments to change the VLAN. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

or:

```
vlan tset <vlanid> new-name <new vlanid>
```

<vlanid> Specifies the VLAN IDs to change the corresponding VLANs. Specify a single VLAN ID or a comma-separated list of VLAN IDs.

new-name Specifies that the name of the VLAN is being changed.

<new vlanid> Specifies the new name to be given the corresponding VLAN.

In the following example, segments 1.4 and 1.5 have been added to VLAN 1:

```
31:PowerHub:ip# vlan show
1               1.2,1.3
32:PowerHub:ip# vlan tset 1 seglist 1.4-1.5
33:PowerHub:ip# vlan show
1               1.2,1.3,1.4,1.5
```

In the following example, VLAN 1 has been changed to VLAN 2:

```
31:PowerHub:ip# vlan show
1               1.2,1.3,1.4,1.5
32:PowerHub:ip# vlan tset 1 new-name 2
33:PowerHub:ip# vlan show
2              1.2,1.3,1.4,1.5
```

15.3.7.2 Deleting a Configured VLAN

To delete a configured VLAN, issue the **`vlan delete`** command. The syntax for this command is:

```
vlan del[ete]  <vlanid>
```

<vlanid> Specifies the VLAN IDs to delete from the corresponding VLANs. Specify a single VLAN ID or a comma-separated list of VLAN IDs.

15.3.8 Allocating Memory for Additional IP Routes

Before the `ip` subsystem commands can be used, memory must be allocated memory for the subsystem by issuing the **`addmem`** command. Memory allocation increases the capacity of the IP route table. Additional memory can be specified in terms of IP routes. The increment is 1K routes. The following example shows the results of this command:

```
1:PowerHub:ip# addmem
IPR: Routing Table is now : 1 K
2:PowerHub:ip#
```

If memory has been allocated for IP routing at the time the configuration is saved with a **`system savecfg`** command, the corresponding `ip` subsystem **`addmem`** command is placed in the configuration file ahead of the other `ip` commands. Thus, it is only necessary to type the **`addmem`** command when the PowerHub is first configured for `ip` routing.

15.3.9 Enabling IP Routing

Since IP routing is by default disabled, IP routing can be enabled after defining the IP interfaces (see Section 15.3), using the following command:

```
enable ip
```

15.4 Showing, Adding, and Deleting IP Routes

This section describes how to display the IP route table and interpret its contents. This section also describes how to manually add and delete static route-table entries. Note that the software makes additions to the IP route table in two basic ways: it “learns” them from a routing protocol (RIP or OSPF) or they are added manually. Learned routes are called “dynamic entries” and user-added routes are called “static entries”.

15.4.1 Showing the IP Route Table

To display the IP route table, issue the following command:

```
route [show] [-c|-r|-s|-o] [d|t] [-a] [-f] [<seglist>] [<ipaddrlist>]
```

[-c|-r|-s|-o] Filters the display according to the type of route:

-c Displays only directly connected entries.

-r Displays only RIP routes.

-s Displays only static routes and directly connected routes.

-o Displays only OSPF routes.

[d|t] Displays additional information, including statistics for packets and bytes. When this argument is specified, the **-f** argument is ignored. **t** displays the total number of routes.

[-a] Displays only active routes.

[-f] Displays routes that are in the DOWN state.

<seglist> Specifies the segments for which route information is to be displayed. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

<ipaddrlist> Specifies the IP address list of segments to display route information. Awildcard character (*) can be used in place of any part of the IP address.

The **-f** argument is used in this example to display selected routes that are DOWN.

```
35:PowerHub:ip# route show -f
Destination Subnet Mask    Gateway    Met  State   RtSrc  Age   Port
 ----------- ------------   ---------  ---  ------  -----  ----  ----
 195.1.1.0   255.255.255.0  ---------   0   Down    Direct ----  None
Total routes: 1 (Direct: 1, Static Routes: 0, RIP routes: 0)
```

For each IP route, the route table shows the following information:

Destination The IP address of the destination host or net.

Subnet Mask The subnet mask used by the destination host or net.

Gateway If the destination is not directly attached to the PowerHub, this field contains the IP address of the gateway (router) through which packets for the destination are to be routed.

Met For entries learned through RIP, this field shows how many hops (routers) away the destination is. For example, if a packet must go through one more router to reach its destination, the metric is 1.

State The state of the route. Possible states are `Up` or `Down`, corresponding to active and inactive.

RtSrc Indicates the source of the routing information:

Direct Indicates that the destination is directly attached to the PowerHub. Such entries are added automatically when the **ip interface add** command is issued.

OExTp1 Indicates that the route was learned through OSPF, from a type-1 External LSA (link-state advertisement). An External LSA indicates that the destination is in another Autonomous System.

OExTp2 Indicates that the route was learned through OSPF, from a type-2 External LSA.

OInter Indicates an inter-area route learned through OSPF. The destination is in an area to which the PowerHub is connected, but the PowerHub is not in the area that contains the destination. OSPF learns the inter-area routes from summary LSAs.

OIntra Indicates an intra-area route learned through OSPF. The destination is in an area that contains the PowerHub.

Static Indicates that the route was manually added using the ip add-route command.

RIP Indicates that the route was learned through RIP.

Age Used only by RIP. Indicates how many seconds have passed since fresh information about this route was received.

Port Lists the segment on which packets for this destination should be forwarded. For directly attached nets, a list of segments can appear, because the PowerHub allows a single net to be used on multiple segments.

15.4.2 Adding IP Routes

The PowerHub stores information about routes in the route table. Entries in the route table are learned dynamically by RIP (as described in *Chapter 17*, *Configuring IP/RIP*), or entries can be configured into the table manually (static entries). Static routes can be assigned for individual hosts or for entire nets.

All nets that have corresponding interface addresses assigned to one or more PowerHub segments are considered to be directly attached.When such interface addresses are assigned by the **`interface add`** command, a corresponding entry is automatically made in the route table. As a result, the routing software automatically routes any incoming IP packet whose destination address is on a directly attached net to the corresponding segment(s). No additional configuration is required.

Additional information is required, however, to route packets to destinations that are not directly attached. In many cases, routers can use RIP to dynamically discover routes that are not directly attached to the hosts and nets. Routes also can be statically assigned, as described in this section. If RIP is not running, routes to non-directly-attached hosts and nets must be assigned statically. To assign the route to be used when forwarding to a host or net, use the **`route add`** command. The syntax for this command is:

```
route add [-s] [-d] <destination> <gw-ipaddr>  <metric> <segment>
```

[-s] Specifies the addition of a strict route.

[-d] Specifies the addition of a route in the DOWN state.

<destination> The IP address of the destination host or net.

<gw-ipaddr> Specifies the IP address of the gateway (router) to which packets destined for the specified host are forwarded. Generally, this gateway is connected to the PowerHub through a net. The net is directly attached to both the gateway and the PowerHub.

<metric> The cost of the route (number of hops to the destination). Generally, the route used is the one with the lowest cost, regardless of whether it is static (added to the route table permanently by the **route add** command) or learned through RIP.

<segment> Specifies the segment onto which a packet is forwarded to reach the specified gateway and the host.

Following is an example of how this command is used.

```
61:PowerHub:ip#rt add 192.9.208.1 255.0.0.0 147.128.128.65 3 5
192.9.200.1 255.0.0.0, 147.128.128.65, 3, 5:  Added-route is active
```

15.4.3 Enabling and Disabling Load Balancing

When load balancing is enabled, the PowerHub, receiving packets from the same source, uses different routes for the incoming packets to reach the PowerHub without any delay. To enable load balancing, issue the following command:

```
load-balance|lb enable|disable
```

enable|disable Specifies whether to enable or disable load balancing. The default is disable.

15.4.4 Enabling Loopback Detection

When loopback detection is enabled, the PowerHub sends a special loopback-detect packet on each outbound segment that has at least one IP address. To enable loop detection on the PowerHub, issue the following command:

```
loop-detection|ld enable|disable
```

enable|disable Specifies whether to enable or disable loop detection.

Following is an example of this command:

```
69:PowerHub:ip# ld enabled
loop-detection:          enabled
```

15.4.4.1 Setting the Looback Detection Time

To set the loopback detection time, issue the following command:

```
loop-detection|ld set time <value>
```

<value> Specifies the time interval in minutes for sending out loopback-detection packets. The default is 10 minutes.

Following is an example of this command:

```
70:PowerHub:ip# ld set time 15
71:PowerHub:ip#
```

15.4.4.2 Displaying the IP Loop Detection Table

To display the IP loop detection table, issue the following command:

```
loop-detection|ld [show]
```

Following is an example of this command:

```
72:PowerHub:ip# ld show
loop-detection:                    enabled
   IP Loop Detection Table:
IP Address         MAC Address            TTL    rport    Segment(s)
---------------    -------------------    ---    -----    -------------
147.128.128.2    08-00-20-08-70-54    16      2         1.3
```

For each IP route, the route table shows the following information:

IP Address The IP address of the outbound segment sending the loopback-detect packet.

MAC Address The Ethernet address of the host.

TTL Specifies how long a packet is allowed to remain in the net before it is dropped. Packets that cannot find or are blocked from their destination nodes are dropped when the TTL expires.

rport Specifies the receiving port of the segments sending the loopback-detect packet.

Segment (s) The segment(s) that sent the loopback-detect packet.

15.4.5 Enabling or Disabling an IP Route

After IP interfaces (see Section 15.3) have been assigned, enable IP routing using the following command:

```
route enable|disable <destination> <gw-ipaddr> <metric> <segment>
```

enable|disable Specifies whether to enabeg or disable IP routing. The default is disable.

<destination> The IP address of the destination host or net.

<gw-ipaddr> Specifies the IP address of the gateway (router) to which packets destined for the specified host are forwarded. Generally, this gateway is connected to the PowerHub through a net. The net is directly attached to both the gateway and the PowerHub.

<metric> For entries learned through RIP, this field shows how many hops (routers) away the destination is. For example, if a packet must go through one more router to reach its destination, the metric is 1.

<segment> Specifies the segments for which to enable or disable the feature. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

15.4.6 Deleting an IP Route

Static routes can be deleted by using the **`route del`** command. To delete a remote, host-specific entry, issue this command:

```
route del <destination> <gw-ipaddr>
```

<destination> The IP address of the destination host or net.

<gw-ipaddr> Specifies the IP address of the gateway.

The **`route del`** command cannot be used to delete learned entries from the route table. The software automatically removes learned entries that remain unused for 180 seconds.

15.5 IP Router Discovery

Based on ICMP to enable hosts attached to multicast or broadcast networks to discover the IP addresses of their neighboring routers, router discovery allows hosts to discover routers automatically through a series of solicitation and advertisement messages. This eliminates the need for the specific configuration of static addresses.

Before a host can send IP datagrams beyond its directly-attached subnet, it must discover the address of at least one operational router on that subnet. Typically, this is accomplished by reading a list of one or more router addresses from a (possibly remote) configuration file at start-up time. On multicast links, some hosts also discover router addresses by listening to routing protocol traffic. Routing discovery on the PowerHub uses a pair of ICMP [10] messages, for use on multicast links. More information about router discovery can be found in RFC 1256. To enable router discovery on the PowerHub, issue the following command:

```
rdm nenable <ipaddr>
```

<ipaddr> Specifies the IP address of the host.

15.5.1 Setting the Advertisement Address

Whether to send out advertise messages to the all-systems multicast address, 224.0.0.1, or to the limited-broadcast address, 255.255.255.255 can be specified. By default, the PowerHub designates the all-systems multicast address. To set the Advertisement Address for Router Discovery on the PowerHub, issue the following command:

```
rdm nset AdvertisementAddress multicast|bro <ipaddr>
```

AdvertisementAddress Specifies the IP destination address to be used for multicast Router Advertisements sent from the interface. The only permissible values are the all-systems multicast address, 224.0.0.1, or the limited-broadcast address, 255.255.255.255.

multicast|bro Specifies advertise messages to all-systems multicast address, 224.0.0.1, or to the limited-broadcast address, 255.255.255.255. The PowerHub default is all-systems multicast, 224.0.0.1.

<ipaddr> Specifies the IP address belonging to the interface from which this message is sent, or 0.

15.5.2 Setting the Advertisement Preference

A Router Advertisement includes a preference level for each advertised router address. When a host must choose a default router address that is, when, for a particular destination, the host has not been redirected or configured to use a specific router address, the host is expected to choose from those router addresses that have the highest preference level. To set the advertisement preference for Router Discovery on the PowerHub, issue the following command:

```
rdm nset preference <preference> <ipaddr>
```

preference Specifies the preference value for Router Discovery is being set.

<preference> Specifies the preference of each Router Address as a default router address, relative to other router addresses on the same subnet. A signed, twos-complement value; higher values mean more preferable.

A 32-bit, signed, twos-complement integer, with higher values meaning more preferable. The minimum value (hex 80000000) is used to indicate that the IP address, even though it may be advertised, is not to be used by neighboring hosts as a default router address.

<ipaddr> Specifies the IP address belonging to the interface from which this message is sent, or 0.

15.5.3 Setting the Advertisement Interval

A Router Advertisement also includes a lifetime field, specifying the maximum length of time that the advertised addresses are to be considered as valid router addresses by hosts, in the absence of further advertisements. This is used to ensure that hosts eventually forget about routers that fail, become unreachable, or stop acting as routers. The default advertising rate is once every 10 minutes, and the default lifetime is 30 minutes. To set the advertisement interval for Router Discovery on the PowerHub, issue the following command:

```
rdm nset interval <time> <ipaddr>
```

interval Specifies the interval value for Router Discovery is being set.

<time> Specifies the time allowed between sending multicast Router Advertisements from the interface.

<ipaddr> Specifies the IP address belonging to the interface from which this message is sent, or 0.

15.5.4 Displaying the Advertisement Interval

To display the Router Discovery table, issue the following command:

```
rdm [show]
```

Here is an example of the results produced by this command:

```
72:PowerHub:ip# rdm show
 -- RDM Configuration --
   IP Address     ADVERTISEMENT       Interval    Preference    this
                     Address                      Level         IP address
---------------  -------------------  --------    ----------    -------------
    200.1.1.1      multicast          10:00           0            yes
    150.1.1.2      multicast          10:00           0            yes
```

15.6 Showing and Configuring the ARP Table

The PowerHub IP routing software maintains an ARP table of IP-to-Ethernet address translations. These translations are used to route packets and, under some circumstances, to generate replies to ARP requests.There are three ways that entries are added to the ARP table:

- When a host uses ARP to request the PowerHub Ethernet address, the host’s IP and Ethernet addresses are recorded (“learned”).
- If a host forwards a packet to a destination through the PowerHub, it can generate an ARP request to learn the destination’s Ethernet address. When a reply to such a request is received, it records the destination’s IP and Ethernet addresses.
- Permanent entries are added using PowerHub commands.

15.6.1 Enabling and Disabling ARP

To enable ARP, issue the following command:

```
arp enable auto-learn
```

auto-learn Indicates enabling auto-learn of incoming packets on the PowerHub. Default is auto-learn enabled.

To disable ARP auto-learning, issue the following command:

```
arp disable auto-learn
```

15.6.2 The ARP Cache

IP route packets are queued for which the ARP table does not contain entries, then sends an ARP request to learn the Ethernet address of the destination device. When the ARP reply is received from the destination device, the queued packet is forwarded. The source node does not need to resend the packet.

15.6.3 Showing the ARP Table

The **arp [show]** command is used to display the current contents of the ARP table. The syntax for this command is:

arp [show] [-r] [-t] [-s] [*<disp-restrictors>*]

[-r] Specifies raw entries with hash indices and displacements.

[-t] Specifies that only the total count of entries is to be displayed.

[-s] Specifies that the ARP entries to be displayed are sorted by the IP address (in increasing order).

[<disp-restrictors>] Specifies segments for which t to display the IP addresses. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

Following are some examples of the use of this command. If no argument is given, then the entire table is displayed, for example:

```
70:PowerHub:ip# arp show
IP Addr        Ethernet Address    Flags          Segment

147.128.128.2  08-00-20-08-70-54    perm publish    6
147.128.128.3  08-00-20-08-85-69                    2
192.9.201.1    02-cf-1f-90-40-23                    12
192.9.201.7    08-00-20-0f-dd-99    perm            10
```

Permanent entries can have a flag, indicating that the entry was added automatically, and a broadcast flag, indicating that the Ethernet address is broadcast or multicast.

An optional IP address can be specified with the **arp [show]** command, in which case only the entry for that address is displayed:

```
70:PowerHub:ip# arp show 147.128.128.2
IP Addr          Ethernet Address      Flags      Segment

147.128.128.2    08-00-20-08-70-54                  6
```

A “wildcard” character (*****) can be used in place of any byte(s) of the IP address, in which case only entries that match that address are displayed.

15.6.4 Clearing the ARP Table

The **arp clear** command is used to clear the ARP table. All learned entries are removed, but static entries (created using the **arp add** command) remain in the table. These must be removed manually using the **arp del** command.

This commandcan be used to help restabilize the network after a host is moved from one segment to another. When there is activity on the network, the cleared entries quickly reappear in the ARP table, and a host that has been moved is relearned on its new segment.

15.6.5 Showing and Changing the ARP Aging Interval

By default, the PowerHub automatically checks learned entries in the ARP table every five minutesto see if they have been used. Each unused entry is marked aged. If an aged entry is used during the next five-minute interval, the aged flag is removed. However, aged entries that remain unused during the second five-minute interval are removed from the ARP table.

The aging interval can be changed or turn off aging using the **arp set age** command. The syntax for this command is:

arp set|show|unset age *<time>*

<time> Specifies (in minutes) a new aging interval or turns aging off. The default is 5 minutes. Set the aging time at a minimum of 1 minute (enter either 60 (seconds) or 1:00). To specify minutes, specify <minutes:seconds>.

NOTE

If ARP aging off is turned off, the ARP table can quickly overflow. Make sure to monitor the table frequently if ARP aging off is turned off.

Following is an example of the use of this command:

```
73:PowerHub:ip# arp set age 30:00
ARP cache aging set to 30 minutes
```

To display the current ARP aging interval, issue the **config show** command.

15.6.6 Adding a Static Entry to the ARP Table

The **arp add** command is used to add a static ARP entry to the ARP table. Static ARP entries are not subject to aging and are not cleared when the ARP table is cleared (using the **arp clear** command). The syntax for this command is:

arp add [-p] *<ipaddr> <ethaddr> <seglist>*

[-p] If this argument is present, then the IP routing software replies directly to ARP requests for this entry. Note that this facility is provided only for permanent, not learned, entries in the ARP table.

<ipaddr> Specifies the IP address to be translated.

<ethaddr> Specifies the Ethernet address corresponding to the given IP address.

<seglist> Specifies the segments to which packets sent to the IP address specified by <*ethaddr*> are forwarded. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. IIf **all** is specified, the ARP entry is added to all segments.

Some examples of using the **arp add** command are shown below:

```
75:PowerHub:ip# arp add -p 147.128.128.8 08-00-02-03-04-05 1.4
Added/changed:
        IP address: 147.128.128.8
        Ethernet address: 08:00:02:03:04:05
        Flags: PERMANENT  PUBLISH
Segments: none
```

Permanent and published entries are flagged in the ARP table:

```
77:PowerHub:ip# at
IP Addr        Ethernet Address   Flags          Segments
147.128.128.1  08:00:02:03:04:05  perm publish   1.1
```

15.6.7 Deleting a Static Entry from the ARP Table

The **arp delete** command is used to delete static or dynamically learned ARP entries from the ARP table. The syntax for this command is:

```
arp delete <ipaddr>
```

<ipaddr> Specifies the IP address in the ARP entry to delete.

When a host is moved from one segment to another, this command can be used to delete its obsolete learned entry from the ARP table without disturbing any other entries. The network can then relearn the host's new location without being forced to relearn other host locations, as it would if the ARP table was cleared.

15.7 Pinging Other IP Devices

The PowerHub supports the echo facility of the ICMP in two ways:

- Aresponse is generated to any ICMP echo request packet received on any segment.
- An ICMP echo request packet can be sent to any IP address.

ICMP echo requests are commonly used to determine whether devices are reachable on the network. UNIX workstations provide a **ping** command that generates an ICMP echo request to a specified IP address.

When the **ping** command is issued from a workstation, the PowerHub responds and can determine whether the PowerHub is reachable from the workstation. However, depending upon the configuration, the PowerHub might be known by multiple IP addresses. Unless the workstation is directly connected, the IP address specified in the **ping** command can affect the route taken, and therefore the reachability of the PowerHub.

Similarly, the PowerHub itself provides a ping command to generate an ICMP echo request to a specified IP address. The syntax for this command is:

```
ping|pi [-t <timeout>] [-size <size>] <ipaddr>
```

[-t <timeout>] Specifies how many seconds the PowerHub waits for a response from the specified device. The default is **5** seconds.

[-size <size>] Specifies the packet length. Specify any length from **64** through **1472** bytes. The default is **64** bytes.

<ipaddr> Specifies the IP address of the distant device.

Here are some examples of the use of this command.

```
83:PowerHub:ip# ping 147.128.128.8
147.128.128.8 is alive
84:PowerHub:ip# ping 147.128.128.15
No response from 147.128.128.15
```

The **`ping`** command normally waits 5 seconds for the specified host to respond before timing out. However, a shorter or longer time-out can be specified, as shown in the following example. In this example, a one-second delay is specified.

```
85:PowerHub:ip# ping 147.128.128.8 1
No response from 147.128.128.8
```

15.8 IP Helper

This section describes how to use the IP Helper feature. IP Helper is an enhancement to the `ip` subsystem that assists client stations on one network segment in communicating with servers on another network segment when the two segments are connected by a PowerHub. This includes situations where one switch, as a client station, needs to boot from a server from which it is separated by another switch.

By default, the IP Helper feature is configured to help packets destined for any of the following standard UDP ports:

- BootP client packets (port 68).
- BootP server packets (port 67).
- Domain Name System (port 53).
- IEN-116 Name Server (port 42).
- NetBIOS Datagram Server (port 138).
- NetBIOS Name Server (port 137).
- TACACS service (port 98).
- TFTP (port 69).
- Time service (port 37).

If it is necessary to add a UDP port to this list, use the **`helper add -d`** command. (See Section 15.8.2.)

15.8.1 How IP Helper Works

When a client sends a broadcast packet addressed to a server that is directly connected to the client, the server:

- Receives the limited broadcast IP packet sent out by the client.
- Uses the client's Ethernet address to look up its corresponding IP address.
- Sends a unicast packet in reply.

This also is true if the client and server are on different segments, but the segments are defined as part of the same VLAN. In this case, the packets are bridged.

However, if the client and server are on different segments separated by a router (gateway), the client’s broadcast packet never reaches the server. If the intervening router is a PowerHub, the IP Helper facility on that PowerHub can be used to tell it where to forward UDP packets sent by the client.

To use IP Helper to help a client reach its server, assign the server’s IP address as an IP Helper address to the PowerHub segment connected to the client. When this segment receives a UDP packet from the client, it forwards the packet to the node that has the IP address corresponding to the PowerHub segment’s IP Helper address.

For the UDP packet to be successfully forwarded, the following criteria must be met:

- The packet must be received on a segment where an IP Helper address is configured.
- The destination UDP port must be in the UDP-helper Port Table on the router. See RFC 1542 for more information.

Because BootP packets (used for netbooting) are UDP packets, IP Helper makes netbooting possible when the client switch and server are separated by a router. Similarly, it facilitates netbooting of diskless workstations.

NOTE

IP Helper does not affect the forwarding of limited-broadcast packets in a virtual LAN environment. The same packet can be forwarded to multiple segments that are on the same virtual LAN.

15.8.2 Using IP Helper

Before you can use IP Helper:

- The PowerHub switch must be configured as an IP router. (See Section 15.3.9.)
- An IP Helper address must be assigned to the segment which connects to the diskless workstation or other device that is being helped. The IP Helper address is the address of the desired server on the network.

To display helper configuration on an IP segment, issue the following command:

config [show] helper

15.8.2.1 Adding an IP Helper Address

To add an IP Helper address to a segment, issue the following command:

```
helper add <IPaddr> [<UDPportlist>] <seglist>
helper add   -d
```

<IPaddr> Specifies the helper address. Specify the IP address of the server as the helper address.

[<UDPportlist>] Specifies any of the standard UDP ports available by default.

<seglist> Specifies the segments on which to add an IP address. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. If **all** is specified, then the IP address is assigned to all valid segments.

-d Displays the contents of the default UDP portlist. Allows you to specify additional default UDP ports.

Following is an example of the use of this command.

```
11:PowerHub:ip# helper add 147.128.42.37 1.4
Helper address 147.128.42.37: added.
```

An IP Helper address is added to segment 1.4 on the router. When used, this IP Helper address routes UDP packets received on segment 1.4 to IP address 147.128.42.37.

Multiple IP Helper addresses can be assigned to a single segment, or multiple segments to a single IP Helper address. Assigning multiple IP Helper addresses to a single segment provides redundancy when multiple servers are used.

15.8.2.2 Deleting an IP Helper Address

To delete an IP Helper address, issue the **helper delete** command. The syntax for this command is:

helper delete *<IPaddr> <UDPportlist>*|default[s]|all *<seglist>*

<IPaddr> Specifies the helper address. Specify the IP address of the server as the helper address.

<UDPportlist>|default[s]|all Specifies the type of UDP port being deleted. If **all** is specified, all UDP ports, including the default ports are deleted.

<seglist> Specifies the segment(s) which connect the router to the client PowerHub. If **all** is specified, all entries assigned to the specified IP address are deleted.

Following is an example of the use of this command.

```
16:PowerHub:ip# helper delete 2.2.2.2 1.2
2.2.2.2:138 (netbios-dgm), port 1.2 :deleted
2.2.2.2:138 (netbios-ns), port 1.2 :deleted
2.2.2.2:138 (tacnews), port 1.2 :deleted
2.2.2.2:138 (tftp), port 1.2 :deleted
2.2.2.2:138 (dns), port 1.2 :deleted
2.2.2.2:138 (name), port 1.2 :deleted
2.2.2.2:138 (time), port 1.2 :deleted
```

15.8.2.3 Displaying Statistics and the UDP Table

To display current statistics for an IP Helper address defined for a segment, issue the helper show command. A table is displayed listing the segment, helper address, the number of packets helped, and the number of packets dropped. The syntax for this command is:

helper show [-p|-s]
helper show -d

[-p|-s] Sorts the IP Helper table by UDP port **-p**, or by segment number **-s**.

-d Displays the contents of the default UDP portlist. Allows additional default UDP ports to be specified.

Following is an example of the use of this command.

```
11:PowerHub:ip# helper show
Helper IP      UDP port Segment   Helped   Reverse  Dropped
-------------  -------- --------  -------  -------  ---------
147.128.48.37   37 time   1.4        0         0      0
```

The table in this example shows that during the current session, IP Helper address 147.128.48.37 has helped four UDP packets (perhaps BOOTP packets) find their IP destinations. The table also shows that one UDP packet was dropped. Note that the **helper show** command lists statistics only for those UDP packets that the PowerHub tried to help. UDP packets can be dropped for any of the following reasons:

- The helping PowerHub does not have a route to the destination address in the UDP packet.
- The helping PowerHub runs out of resources to redirect the packet.

In addition, for BOOTP packets only, the following conditions can cause the helping PowerHub to drop the packet:

- The hop count in the packet has been exceeded.
- A gateway has already helped the packet. (A bit in the packet is set when the packet is helped.)

15.8.2.4 Deleting Default UDP Entries

To delete default UDP entries, issue the **helper delete** command. The syntax for this command is:

helper delete -d *<UDP ports to remove>*

-d Displays the contents of the default UDP portlist. Allows additional default UDP ports to be deleted.

15.8.2.5 Clearing Statistics

To clear the IP Helper statistics, issue the **stats clear helper** command. Following is an example of the use of this command.

```
12:PowerHub:ip# stats clear helper
IP helper table stats are cleared.
```

15.8.2.6 Deleting an IP Helper Address

To delete an IP Helper address, issue the del-helper command. The syntax for this command is:

helper delete <***IPaddr>* [*<UDPportlist>*|default[s]|all *<seglist>***

helper delete -d *<UDP ports to remove>*

<IPaddr> Specifies the helper address. Specify the IP address of the server as the helper address.

[<UDPportlist>|default[s]|all Displays the contents of the default UDP portlist. If the UDP port list is set to **all**, then the IP address is deleted from all valid segments.

<seglist> Specifies the segment(s) which connect the router to the client switch. If **all** is specified, all entries assigned to the specified IP address are deleted.

-d Displays the contents of the default UDP portlist. Allows additional default UDP ports to delete to be specified.

If both *<seglist>* and *<IPaddr>* are specified as **all**, all IP Helper definitions on the router are deleted.

15.8.2.7 Adding an IP Helper Gateway IP Address

To add an IP Helper gateway IP address to a segment, issue the following command:

helper add -g *<GwIP-Add> <seglist>*

<GWIP-Add> Specifies the defined IP address to be used as the gateway address while helping a bootp/DHCP packet. A maximum of 10 gateway addresses can be configured per segment. When more than one gateway address is configured for a segment, all the gateway addresses are used sequentially.

<seglist> Specifies the segments on which to add a gateway IP address. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

-g Specifies gateway address configuration of the available gateway IP addresses.

Following is an example of the use of this command.

```
11:PowerHub:ip# helper add -g 147.128.42.37 1.4
Helper address 147.128.42.37: added.
```

A gateway IP Helper address is added to segment 1.4 on the router. When used, this IP address is used as the gateway address when a packet being helped is routed.

Multiple IP Helper gateway addresses can be assigned to a single segment, or multiple segments to a single IP Helper gateway address.

15.8.2.8 Deleting an IP Helper Gateway Address

To delete an IP Helper address, issue the **`helper delete`** command. The syntax for this command is:

`helper delete -g` *<GwIP-Add> <seglist>*

<GwIP-Add> Specifies the defined IP address used as the gateway address to help a bootp / DHCP packet to delete.

<seglist> Specifies the segments on which to delete a gateway IP address. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

-g The contents of the available gateway IP addresses to delete.

15.8.2.9 Displaying IP Helper Gateway Addresses

To display an IP Helper address, issue the **`helper show`** command. The syntax for this command is:

`helper show -g`

-g Displays the contents of the available gateway IP addresses to display.

Following is an example of the use of this command.

```
11:PowerHub:ip# helper show
Segment   Configured GW_Addrs
--------  --------------------
  1.4     147.128.42.37
```

15.8.3 Setting the Time-To-Live Parameter

The time-to-live (TTL) parameter specifies how long a packet is allowed to remain in the net before it is dropped. Packets that cannot find or are blocked from their destination nodes are dropped when the TTL expires. To change the default TTL, issue the following command:

`ipdefaultttl|ittl set` <***`value`*****>**

<value> Specifies the new TTL time in hops. Specify a number between 1 and 255. The default is 16 hops.

To display the TTL value for outgoing IP packets, issue the following command:

```
ipdefaultttl|ittl [show]
```

15.8.4 Enabling and Disabling ICMP Redirect Messages

Use the **send-icmp-redirect**command to enable or disable sending ICMP redirect messages by the PowerHub. In networks that use multiple routers, ICMP redirects messages from routers of alternative routes to segments connected to the routers. Normally, this feature helps optimize routing throughput by ensuring that routers are informed of the most efficient paths to the segments on the network.

The PowerHub works well when it receives ICMP redirect messages; however, some other switches do not work well in environments in which these messages are used. If the network contains switches that do not work well when they receive ICMP redirect messages, sending of these messages can be disabled on the PowerHub.

```
enable|disable send-icmp-redirect|sir
```

enable|disable Specifies whether ICMP redirect messages are to be enabled or disabled. The default is **enl** (enabled).

15.8.5 Enabling or Disabling Source-Route Filtering

Use the **fwd-pkts-with-srcrt-options** command to disable the source-route feature and strengthen the "firewall" protecting the network from outside users.

IP packets that contain the loose-source-route or the strict-source-route option are forwarded by default. The source-route options are intended to help forwarding of IP packets. When a packet containing a source-route option is forwarded, the packet can appear to receiving devices as though it originated from the device that forwarded it. As a result, these devices are more likely to accept the forwarded packets, rather than filter them.

Disabling the source-route feature prevents outside users from using and exploiting the source-route contained in packets to gain access to the network. The syntax for this command is:

```
enable|disable fwd-pkts-with-srcrt-option|fps
```

enable|disable Specifies whether source-route filtering is to be enabled or disabled. The default is **enl** (enabled).

NOTE

For additional information about IP filtering, see the *PowerHub Filters Reference Manual.*

15.8.6 Enabling or Disabling Network-Broadcast Forwarding

By default, the PowerHub forwards broadcast packets onto subnets attached to the PowerHub. A network broadcast packet is a packet containing either all zeros or all ones in the host portion of the address. For example: 1.120.255.255, 192.9.200.0, and 10.255.255.255 all are network broadcast packets. The way the software handles broadcast packets differs depending upon how they are received and the destination address specified in the packets.

The PowerHub can be forced to forward or drop IP network-broadcast packets sent to subnetted interfaces, by enabling or disabling bridge-net-broadcast and route-net-broadcast:

- The bridge-net-broadcast state affects network-broadcast packets received in Ethernet-broadcast packets. If bridge-net-broadcast is enabled, these packets are forwarded. If bridge-net-broadcast is disabled, these packets are dropped.
- The route-net-broadcast state affects network-broadcast packets received in Ethernet-unicast packets. If route-net-bcast is enabled, these packets are forwarded. If route-net-bcast is disabled, these packets are dropped.

The bridge-net-bcast and route-net-bcast states are completely independent of each other. Both or only one can be enabled, or both disabled, depending upon the level of broadcast traffic allowed for subnetted interfaces.

IP network-broadcast and IP subnet-broadcast packets can be encapsulated in one of the following types of packets:

- Ethernet-broadcast packets. These packets contain (encapsulate) IP subnet-broadcast packets or IP network-broadcast packets. Ethernet-broadcast packets contain the Ethernet broadcast address (ff-ff-ff-ff-ff-ff) in the destination address field and are received by the PowerHub switch from a directly-attached node.
- Ethernet-unicast packets. These packets contain the PowerHub switch's Ethernet address in the destination field. Like Ethernet-broadcast packets, Ethernet-unicast packets can contain (encapsulate) IP subnet-broadcast packets or IP network-broadcast packets. However, unlike Ethernet-broadcast packets, Ethernet-unicast packets are received by the PowerHub switch from another router.

Forwarding of the following types of IP network-broadcasts can be selectively enabled or disabled:

- IP network-broadcasts sent from a node directly-attached to the switch and addressed to a subnetted interface configured on the switch. If bridge-net-bcast is enabled, the packets are bridged to all segments belonging to all subnets in the destination network. If bridge-net-bcast is disabled, the packets are dropped.

- IP network-broadcasts sent from another router and addressed to a subnetted interface configured on the switch. If route-net-bcast is enabled, the packets are routed to all segments belonging to all subnets in the destination network. If route-net-bcast is disabled, the packets are dropped.

Neither the bridge-net-bcast state nor the route-net-bcast state has any effect on IP subnet-broadcast packets or broadcast packets sent to interfaces that are not subnetted:

- If the interface is subnetted and the received packet is a subnet-broadcast, the packet is unconditionally bridged to all the segments belonging to the same subnet.
- If the interface is not subnetted and the received packet is a network broadcast, the packet is unconditionally bridged to all the segments belonging to the same network.
- If the interface is subnetted, and the received packet is a subnet-broadcast, the packet is unconditionally forwarded (routed) to all the segments in the subnet.
- If the interface is not subnetted, and the received packet is a net-broadcast packet, the packet is unconditionally forwarded (routed) to all segments in the network.

15.8.6.1 Disabling Bridging of Net Broadcasts

To prevent forwarding of IP network-broadcast packets from directly-attached nodes to subnetted interfaces on the PowerHub, issue the following command:

```
disable bridge-net-broadcast|bnb
```

After this command is issued, network-broadcast packets encapsulated in Ethernet-broadcast packets are still received internally, if applicable, but dropped without being forwarded to the destination subnets. Network-broadcast packets received in Ethernet-unicast packets are not affected.

To re-enable the software to forward network-broadcast packets received in Ethernet-broadcast packets and addressed to subnetted interfaces, issue the following command:

```
enable bridge-net-broadcast|bnb
```

15.8.6.2 Disabling Routing of Net Broadcasts

To prevent the forwarding of IP network-broadcast packets from other routers to subnetted interfaces attached to the PowerHub, issue the following command:

```
disable route-net-broadcast|bnb
```

After this command is issued, network-broadcast packets encapsulated in Ethernet-unicast packets are still received internally, if applicable, but dropped without being forwarded to the destination subnets. Network-broadcast packets received in Ethernet-broadcast packets are not affected.

To re-enable the software to forward network-broadcast packets received in Ethernet-unicast packets and addressed to subnetted interfaces, issue the following command:

```
enable route-net-broadcast|bnb
```

15.8.7 Enabling Proxy ARP

The PowerHub supports proxy ARP (RFC 1027), a well-defined mechanism in the TCP/IP protocol suite. Using proxy ARP, a router can respond to an ARP request with its own Ethernet address if it knows a route (or default route) to the destination network or subnet on which the requested address resides.

Without proxy ARP, the requesting host needs to have knowledge of its own network, as well as the destination network and the subnet mask, so that it can ARP the destination directly if it is on the same net or ARP the PowerHub (or other gateway) if the destination is on a different net.

Use the proxy-arp command to enable or disable the proxy ARP feature for all segments or a specific list of segments. The syntax for this command is:

```
proxy-arp penable|pdisable [<seglist>]
```

<seglist> Specifies the segments for which enable or disable the feature. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

penable|pdisable Specifies whether to enable or disable the feature. The default is disable (for all segments).

If a *<seglist>* or **penable|pdisable** is not specified, the current status (enabled or disabled) of the proxy ARP feature is shown.

15.8.7.1 Displaying the Proxy ARP Table

To display the Proxy ARP table, issue the following command:

```
proxy-arp [show]
```

Following is an example of the results produced by this command:

```
71:PowerHub:ip# proxy-arp show
Segment 2.1: disabled
Segment 2.2: enabled
Segment 2.3: enabled
Segment 2.4: enabled
Segment 2.5: disabled
Segment 2.6: enabled
Segment 2.7: disabled
Segment 2.8: disabled
```

15.9 Showing and Clearing Statistics

The `ip` subsystem maintains statistics on ARP, ICMP, and general IP packets. These statistics are a superset of the corresponding statistics provided in the SNMP MIB. Use the **`stats [show]`** command to display statistics. The syntax for this command is:

```
stats [show] [-t] [arp|icmp|ip|helper]
```

-t Specifies to display all statistics collected since the software was rebooted, rather than just the statistics collected since the last time the **`stats clear`** command was issued.

[arp|icmp|ip|helper Specifies the type of packet protocol to display statistics.

The PowerHub maintains two copies of each IP statistics counter (and similarly for ICMP and ARP packets):

- Count since last clear.
- Count since last switch reset.

Both counters are updated when the corresponding events occur, but the **`stats clear`** command clears only the count since last clear. To display the count since last reset, use the **`-t`** option with the **`stats`** command.

Following are some examples of the information displayed by the **`stats`** command. Notice that the first line in each example informs that statistics since the last statistics clear are being displayed, rather than total statistics accumulated since the last reboot.

```
IP statistics: count since last stats clear

            Number of Cache Flushes:  1
```

As shown in this example, the IP statistics are organized according to incoming packets and outgoing packets. In addition to totals for packets received, sent, and forwarded, the **`stats ip`** display lists statistics for many of the types of IP routing errors that can occur in a network.

In the following example, ARP statistics are displayed.

```
74:PowerHub:ip# stats arp
ARP statistics: count since last stats clear
ARP Packet Statistics:
        Requests received:        38
        Replies received:         25
        Invalid opcodes received: 0
        Requests sent:            226
        Replies sent:             36 (0 proxies)
```

Here is an example of the ICMP statistics displayed by the **`stats`** command.

```
74:PowerHub:ip# stats icmp
ICMP statistics: Count Since last stats clear
Messages received:          0   Errors received:          0
Dest unreach msgs rcvd:     0   TTL expired msgs rcvd:    0
Param prob msgs rcvd:       0   Src quench msgs rcvd:     0
Redirect msgs rcvd:         0   Echo request msgs rcvd:   0
Echo reply msgs rcvd:       0   Timstamp req msgs rcvd:   0
Timstamp rpl msgs rcvd:     0   AddrMask req msgs rcvd:   0
AddrMask rpl msgs rcvd:     0
Messages sent:            200   Errors sent:              0
Dest unreach msgs sent:   200   TTL expired msgs sent:    0
Param prob msgs sent:       0   Src quench msgs sent:     0
Redirect msgs sent:         0   Echo request msgs sent:   0
Echo reply msgs sent:       0   Timstamp req msgs sent:   0
Timstamp rpl msgs sent:     0   AddrMask req msgs sent:   0
AddrMask rpl msgs sent:     0
```

15.9.1 Clearing Statistics

The **`stats clear`** command is used to clear statistics. The syntax for this command is:

`stats clear [arp|icmp|ip|helper|all]`

[arp|icmp|ip|helper Specifies the type of packet protocol for which to clear statistics.

15.10Showing or Clearing the IP Route Cache

The IP routing software maintains a route cache containing translation information for the destination hosts. This information is frequently updated based upon incoming packets on each segment. The route cache can be used to determine which hosts are most frequently used.

15.10.1 Displaying the Route Cache

The **cache show** command is used to display the route cache. The syntax for this command is:

cache show [*<seglist>***]**

[<seglist>] Specifies the segments for which to display the route cache. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

15.10.2 Flushing the Route Cache

The route cache can be cleared using the **cache clear** command. The **cache clear** command removes all entries from the route cache for some or all segments.

After the cache is flushed, new entries are added using the cache's usual most-recently-used algorithm. If a subsequent **cache show** command is issued, fresh entries are displayed.

CHAPTER 16 Configuring IP Multicast

This chapter describes the IP Multicast commands and how to use them to define IP interfaces on a PowerHub as end stations for IP Multicasting. Unlike IP broadcasting, which sends packets to all destinations, or IP unicasting, which sends packets to a single destination, IP Multicasting addresses and delivers packets to a specific subset of destinations.

Using the `ip/mcast` subsystem, IP Multicasting can be set up and used with video conferencing and other multicast applications. Using `ip/mcast` commands:

- Show the IP Multicast configuration
- Add, show, and delete IP Multicast interfaces
- Add and delete IP Multicast tunnels
- Enable IP Multicast routing
- Show and clear the IP Multicast route table
- Show and clear the IP Multicast route cache
- Show and clear IP Multicast statistics
- Enable multicast-aware bridging (for systems that perform IP Multicast routing on VLANs)

16.1 Accessing the IP Multicast Subsystem

To access the `ip/mcast` subsystem, issue the following command at the runtime command prompt:

```
ip/mcast
```

16.1.1 Allocating Memory

Before using the `ip/mcast` subsystem, memory must be allocated by issuing the **getmem** command, as shown in the following example:

```
1:PowerHub:ip/mcast# getmem
Memory allocated for IP Multicast.
2:PowerHub:ip/mcast#
```

If memory has been allocated for IP Multicast at the time the configuration is saved with the **savecfg** command, the corresponding **getmem** command is placed in the configuration file ahead of other IP Multicast configuration commands. Thus, it is only necessary to type the **getmem** command when first configuring the PowerHub for IP Multicast routing.

NOTE

FORE Systems recommends that memory for the `ip/mcast` subsystem be allocated immediately after booting to ensure that the memory requested is available. For more information, refer to the *PowerHub Hardware Reference Manual*.

Memory cannot be deallocated. To free allocated memory, make sure the configuration file does not contain a **getmem** command, then reboot the software.

16.1.2 Enabling Pruning

To enable or disable pruning in the IP Multicast subsystem, issue the following command:

```
pruning enable|disable
```

Following are the results of this command:

```
311:PowerHub:ip/mcast# pruning enable
IP Multicast pruning enabled.
312:PowerHub:ip/mcast#
```

16.2 Showing the IP Multicast Configuration

The current IP Multicast configuration can be displayed by issuing the **config show** command. Following is an example of the information shown by the **config show** command.

```
44:PowerHub:ip/mcast# show config
   IP Multicast Forwarding: disabled
   Multicast Aware Bridging in a VLAN: disabled
   IPM Pruning: enabled
   Max Routing Entries allocated: 2k

Port State for Multicast Traffic:
Segment  2.1:  Disabled ***
Segment  2.2:  Enabled
Segment  2.3:  Enabled
Segment  2.4:  Enabled
```

In this example, the display produced by the **show config** command shows:

- IP Multicast forwarding is enabled.
- Multicast Aware Bridging in a VLAN is disabled.
- IPM pruning is enabled.
- Maximum routing entries allocated is 2k.
- IP Multicast traffic is enabled on all segments except 2.1.

16.2.1 IP Considerations

IP Multicast routing works whether IP forwarding is enabled or disabled. In this respect, the PowerHub implementation is similar to mrouted, which allows multicast routing on a UNIX workstation even if it is not routing regular IP traffic.

NOTE

IP Multicast routing must be enabled even if the PowerHub is configured to have the same subnet on all the segments. The IP Multicast routing code bridges packets intelligently based on reception of membership reports. IP Multicast traffic is restricted to those networks that have listening hosts.

The virtual interface table used for IP Multicast routing is associated closely with the IP interface table. When a virtual interface is added, appropriate information is automatically copied from the IP interface table.

The PowerHub updates the segment list in the virtual interface table whenever adding or deleting a segment in an IP interface entry. When an IP interface entry is deleted, all the IP Multicast virtual interfaces that match the deleted entry's address are deleted.

16.2.2 Displaying IP Multicast Groups

The **`multicast-groups show`** command is used to list the IP Multicast group addresses currently known to the PowerHub (local router). Following is an example of the display produced by this command.

```
35:PowerHub:ip/mcast# multicast-groups show
Virtual I\F- : Locaddr: 147.128.70.30  RmtAddr :----, type: Physical
Groups: 224.2.138.32   Segs: 2.1

Virtual  I/F-  Locaddr: ---  RmtAddr:147.128.90.33, type: Tunnel
```

This table contains the list of IP Multicast groups for each virtual interface and contains the following information:

Locaddr Displays additional statistics, including the number of packets and octets transmitted to and received from the net by each interface.

RmtAddr Lists the IP address of a remotely attached IP Multicast neighbor. This applies only to tunnels, in which the PowerHub and the other end of the virtual interface are separated by gateways.

type Lists the type of IP Multicast interface. Valid types are `Physical` and `Tunnel`.

Groups Lists the IP Multicast groups. The group IP address and the PowerHub segment(s) on which membership reports for that group were received are listed for each group.

16.2.3 Displaying IP Multicast Neighbors

The **`neighbors show`** command is used to list all the neighboring routers currently known. Following is an example of the display produced by this command.

```
35:PowerHub:ip/mcast# neighbors show
Virtual I\F- :Locaddr: 147.128.128.99 RmtAddr :----,type:Physical,
Neighbors: 147.128.128.30   (25 sec)   147.128.100.2  (40 sec)

Virtual  I/F-  Locaddr:  147.128.128.99 Rmtaddr---,type:Tunnel,
Neighbors: 130.1.5.1   (35 sec)
```

This display contains a list of neighboring routers for each virtual interface and contains the following information:

Locaddr Lists the IP address of a directly-attached IP Multicast neighbor. This applies only to physical interfaces, in which the PowerHub and the other end of virtual interface are directly attached.

RmtAddr Lists the IP address of a remotely attached IP Multicast neighbor. This applies only to tunnels, in which the PowerHub and the other end of the virtual interface are separated by gateways.

type Lists the type of IP Multicast interface. Valid types are `Physical` and `Tunnel`.

Neighbors Lists the IP Multicast neighbors. The router's IP address and the number of seconds elapsed since the last routing update was received from this neighbor is listed for each neighbor.

16.3 Configuring and Displaying IP Multicast Interfaces

A physical interface allows two directly connected PowerHubs (local routers) to communicate with each other. To define a physical interface, use the **interface add** command. The syntax for the command is:

it|interface add *<ipaddr>* [met[ric]*<metric>*] [thresh[old]*<thresh>*]

<ipaddr> IP address on the local switch, written in dotted-decimal notation. The address must be present in the IP interface table.

[met[ric]<metric>] Specifies any additional cost (measured in hops to the destination) of using the interface. The cost range is from 1 through 31. The default is **1**.

[thresh[old]<thresh>] Specifies the minimum time-to-live (TTL) value that an IP Multicast packet must have before it is forwarded over this interface.

This parameter can restrict the types of IP Multicast traffic that go out on a network. The default is **1**.

Following is an example of the use of the **interface add** command:

```
32:PowerHub:ip/mcast# interface add 192.10.30.33
Okay
33:PowerHub:ip/mcast#
```

16.3.1 Displaying the Interface Table

The **interface [show]** command can be used to display a list of configured virtual interfaces. The display includes both physical interfaces and tunnels. The syntax for this command is:

it|interface [show] [*<disprestrictors>*]

[<disprestrictors>] Specifies segments for which to display the IP addresses. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

Following is an example of the interface table displayed by the **interface [show]** command.

```
33:PowerHub:ip/mcast# it
       IP Multicast Routing: virtual interface table:
       LocalAddress  RemoteAddress Type SrcRt Metrc Thrsh State Segments
       ------------- ------------- ---- ----- ----- ----- ----- --------
       147.128.70.30 ------------- Phy  ----- 1     1     Up    2.4,1.8
       147.128.128.99------------- Phy  ----- 1     1     Up    2.3
       147.128.33.5  130.1.5.1     Tunl No    1     6     Up    2.4,1.8
       147.128.30.30 ------------- Phy  ----- 1     1     Up    1.5
       147.128.33.5  192.9.200.21  Tunl Yes   1     6     Up    1.6
       147.128.33.5  ------------- Phy  ----- 1     1     Up    1.2
```

This display contains information about four physical interfaces and two tunnels. The tunnel to destination 130.1.5.1 is an encapsulation tunnel. The tunnel to destination 192.9.200.21 is a source-route tunnel.

Local Address and Remote Address Identifies the two ends of a tunnel. The local address corresponds to the configured address for a physical interface.

Type Identifies whether the virtual interface is either a tunnel or a physical interface.

SrcRt Identifies the type of tunnel. Yes in this column indicates that the tunnel is a source-route tunnel. No in this column indicates that the tunnel is an encapsulation tunnel.

Metrc Lists the cost (in hops) of the interface.

Thrsh Lists the threshold value for the interface.

State Indicates the state of the interface. Up indicates the interface is active. Down indicates the interface is inactive. The interface is DOWN when a segment from the bridging subsystem is disabled, or if disabled by the automatic segment-state detection mechanism. See your *PowerHub Hardware Reference Manual* for further information on automatic segment-state detection.

Ports Lists the segments to which the listed virtual interface is assigned.

16.3.2 Deleting a Physical Interface

The **`interface del`** command is used to delete a physical interface.

NOTE When a physical interface is deleted, corresponding tunnels are not deleted. To delete a tunnel, use the **`tunnel del`** command.

The syntax for the **`interface delete`** command is:

```
it|interface delete <ipaddr>|all
```

<ipaddr>|all Specifies the IP address of the physical interface to be deleted.

If **`all`** is specified, all physical interfaces (excluding the tunnels) are deleted.

16.3.3 Deleting a Tunnel

The **`tunnel del`** command is used to delete a virtual interface that maps to a tunnel. The syntax for this command is:

```
tunnel del (loc[al]<local-addr> rem[ote]<remote-addr> )|all
```

loc[al]<localaddr> Specifies the IP address of the PowerHub (the local end of the tunnel).

rem[ote]<remoteaddr> Specifies the IP address of the router at the remote end of the tunnel.

To delete all IP Multicast tunnels, issue the following command:

```
tunnel del all
```

16.4 Configuring and Displaying Tunnels

A tunnel is a type of virtual interface that allows the PowerHub (local router) to communicate with a remotely attached router.

16.4.1 Adding a Tunnel

The **tunnel add** command is used to define a tunnel. The syntax for this command is:

```
tunnel add [-s] loc[al]<local-addr> rem[ote]<remote-addr>
        [met[ric]<mv>] [thresh[old]<tv>]
```

[-s] Specifies that the tunnel is a source-route tunnel, rather than an encapsulation tunnel.

If **-s** is not specified, this command automatically configures the tunnel as an encapsulation tunnel.

loc[al]<localaddr> Specifies the IP address of the local PowerHub. The address must be one of the configured IP addresses listed in the IP interface table.

rem[ote]<remoteaddr> Specifies the IP address of the router at the other end of the tunnel.

[met[ric]<mv> Specifies an additional cost (extra hops to the destination) of using the virtual interface with which this tunnel is associated. Specify a number in the range 1 through 31. The default is **1**.

[thresh[old]<tv> Specifies the minimum time-to-live (TTL) value that an IP Multicast packet must have before it can be forwarded through the tunnel. This parameter restricts IP Multicast datagrams from going out on a network. The default is **1**.

Following is an example of how to add a tunnel. In this example, the **-s** argument is not used, so the software creates an encapsulation tunnel. The default values are accepted for the metric and threshold.

```
34:PowerHub:ip/mcast#tunnel add loc 192.10.30.33 rem 155.10.23.222 met 3 thresh 4
Okay
35:PowerHub:ip/mcast#
```

16.4.2 Deleting a Tunnel

The **tunnel del** command is used to delete a virtual interface that maps to a tunnel. The syntax for this command is:

tunnel del (loc[al]*<local-addr>* rem[ote]*<remote-addr>*)|all

loc[al]<localaddr> Specifies the IP address of the PowerHub (the local end of the tunnel).

rem[ote]<remoteaddr> Specifies the IP address of the router at the remote end of the tunnel.

To delete all IP Multicast tunnels, issue the following command:

tunnel del all

16.5 Enabling IP Multicast Routing

The **enable ipm** command is used to enable IP Multicast forwarding. The syntax for this command is:

enable|disable ipm

enable|disable Specifies whether you are enabling or disabling IP Multicast forwarding. The default is **dis**.

NOTE

If IP Multicast forwarding is enabled immediately after enabling RIP listening, the multicast route updates are not accepted over a tunnel until the IP routing table learns either an entry to the remote end of the tunnel or a default route.

16.5.1 Enabling Multicast Traffic on a Segment

IP Multicast forwarding can be restricted on a segment-by-segment basis. The syntax for the command used to enable or disable IP Multicast traffic on a set of segments is shown below.

`penable|pdisable transmit` *<segment-list>*

penable|pdisable Specifies whether IP Multicast forwarding is to be enabled or disabled. The default is **`penable`**.

<segment-list> Specifies the list of segments on which IP Multicast forwarding is being enabled or disabled. If **`all`** is specified, IP Multicast forwarding is enabled or disabled on all segments.

The first command in the following example uses the **`transmit`** command to enable IP Multicast traffic on segments 2.4. The second command uses the set command to disable IP Multicast traffic on segment 2.2.

```
46:PowerHub:ip/mcast# penable transmit 2.4
Ok
47:PowerHub:ip/mcast# pdisable transmit 2.2
Ok
```

16.6 Configuring and Displaying IP Multicast Routes

The **`route show`** command is used to display a list of IP addresses originating IP Multicast traffic, currently known by the IP Multicast routing software. The syntax for this command is:

`route|rt [show] [-c|-r] [-d] [-t] [-s] [`*`<seglist>`*`] [`*`<ipaddr>`*`]`

[-c|-r] **`-c`** displays directly connected routes only.
`-r` displays Distance Vector Multicast Routing Protocol (DVMRP) routes only.

[-d] Displays the routing table in detail.

[-t] Displays the total number of routes only.

[-s] Displays the output in sorted order.

[<seglist>] Specifies the PowerHub segments for which to display route information.

[<ipaddr>] Specifies the IP address (origin) of the route entries to be displayed.

Following is an example of the display produced by the command.

```
52:PowerHub:ip/mcast# route show
        IP Multicast Routing table:
        Origin         Origin Mask     Gateway          Met  Age  Par.  Seg/Child
        -----------    --------------  -------------    ---  ---  ----  -------
        147.128.70.0   255.255.255.0   --------------   1    ---  ----  4.1
        147.128.128.0  255.255.255.0   --------------   1    ---  ----  2.2
        147.128.90.0   255.255.255.0   --------------   1    ---  ----  5.1,6.1
        129.155.80.0   255.255.240.0   147.128.70.2     3    20   4     2.5,6.2
        150.233.0.0    255.255.0.0     147.128.128.111  5    35   2     4.5,2.6
```

The route table contains the following information:

Origin Lists the IP address of the origin network. An origin is a network that is capable of originating IP Multicast traffic.

Origin Mask Lists the origin mask used on the origin network. An origin mask is the subnet mask of an origin network.

Gateway Lists the IP address of the next-hop router to the origin. This column is not applicable to directly connected entries.

Met Displays the total cost (or metric) of reaching the origin. This metric is the sum of the cost of the next-hop virtual interface and the number of hops or intervening routers (if applicable) used to reach the origin.

Age Shows the time elapsed, in seconds, since a DVMRP route report was last received for this origin. This column is not applicable to directly connected routes.

Parent Shows the segment on which the next-hop router is located for a dynamic route. This column is not applicable to directly connected routes.

For directly-connected routes, the Seg/Children column shows the segments on which the corresponding virtual interface is configured. For a dynamic entry, this column lists the segments on which the IP Multicast packets from this origin are forwarded.

16.6.1 Clearing the Route Table

The **route clear** command is used to flush all dynamically learned entries from the route table. Following is an example of the use of this command.

```
66:PowerHub:ip/mcast# route clear
Okay
```

16.7 Using the IP Route Cache

The IP Multicasting software maintains a route cache containing translation information for the destination hosts. This information is frequently updated based upon incoming packets on each segment. The route cache can be used to determine which hosts are most frequently used. Because the contents of the route cache can change rapidly, successive **cache show** commands can give different results.

16.7.1 Displaying and Clearing the Route Cache

The **cache show** command is used to display the route cache. The syntax for this command is:

```
cache show
```

The route cache can be flushed (cleared) using the **cache clear** command. This command removes all entries from the route cache for some or all segments. After the cache is flushed, new entries are added using the cache's usual most-recently-used algorithm. If a subsequent **cache show** command is issued, fresh entries are displayed.

16.8 Displaying Statistics

The `ip/mcast` subsystem maintains statistics on DVMRP, Internet Group Management Protocol (IGMP) and routed packets. To display statistics, issue the following command:

```
stats [show] [-t] [dvm|igmp|rt|all]
```

[-t] Displays statistics collected subsequent to the last system reset, rather than merely the last time statistics were cleared.

[dvm|igmp|rt|all] Displays the type of packet for which statistics are desired:

dvm Displays DVMRP packet statistics.

igmp Displays IGMP packet statistics.

rt Displays routing packet statistics.

Following is an example of the display produced by the **`stats show dvm`** command, used to display DVMRP statistics.

```
5:PowerHub:ip/mcast# stats show dvm
DVMRP Statistics (count since last stats clear):
Route reports sent:                                  32
Neighbor probes sent:                              0
Neighbor prunes sent:                              0
Neighbor graphs sent:                              0
Neighbor graft_acks sent:                      0
Neighbor responses sent:                  12
Neighbor2 responses sent:                 0

Route reports received:                          211
Neighbor Probes received:                 1
Neighbor prunes received:                      0
Neighbor graphs received:                      0
Neighbor graft_acks received:              0
Neighbor requests received:                  0
Neighbor2 requests received:              33

Rcvd pkts with bad metric:                1
Rcvd pkts with bad orig. mask:            0
Rcvd conflicting route reports:           0
Rcvd truncated route reports:             0
Conflicting routes deleted:               0
Rcvd reports from non neighbor:           5
Rcvd probes from non neighbor             5
Rcvd prunes from non neighbor             5
Rcvd grafts from non neighbor             5
Rcvd graft_acks from non neighbor         0
Rcvd invalid neighbor requests:           0
Rcvd invalid neighbor responses:          0
Rcvd invalid Neighbor2 responses:         0
Rcvd message from non neighbor:           0

No mem to receive packet:                 2
No memory to send packets:                0
```

Following is an example of the display produced by the **`stats show igmp`** command, used to display IGMP statistics.

```
60:PowerHub:ip/mcast# stats show igmp
IGMP Statistics (count since last stats clear):
total packets received:                       551
short packets received:                       2
pkts rcvd with checksum error:                0
total membership queries rcvd:                12
invalid membership queries rcvd:              0
total membership reports rcvd:                333
invalid membership reports rcvd:              0
rcvd packets too big:                           0
rcvd unknown DVMRP message:                   0
rcvd unknown IGMP message:                    0
packets looped back:                             9
no buffer for looping back:                   0
no timers for IP Multicast routing:           0
report not sent - no interface:               0
group timer not started - no I/F:             0
rcvd report from non adj. host:               1
total membership queries sent:                9
total packets sent:                               159
total packets not sent:                       0
no memory to process rcvd pkts:               2
Queue blocks accessed:                        2
Queue blocks released:                        2
Free Queue blocks available:                  2048
```

An example of the display produced by **stats show rt** command, to display routing statistics.

```
59:PowerHub:ip/mcast# stats show rt
Multicast routing statistics (count since last clear):
route cache hits:                                          661
route cache misses:                                  661
route cache flushed:                                 0
route lookups:                                                661
route cache misses:                                  661
source group pair cache lookups:                     11322
source group pair cache misses:                      11322
rcvd msg over invalid tunnel:                        5
no room for tunnel options:                          0
rcvd msg on wrong interface:                         17
packets forwarded:                                               3213
packets dropped:                                                           2448
packets received:                                                         5661
rcvd packet format error:                            0
encapsulated packets rcvd:                           2112
rcvd port not configured:                            0
no route to origin:                                             2448
packets bridged:                                                           1123
packets not bridged:                                 0
no memory to process packets:                        0
```

16.8.1 Clearing Statistics

The **stats clear** command is used to clear statistics for DVMRP, IGMP, or route packets. The syntax for this command is:

```
stats clear [dvm|igmp|rt|all]
```

16.9 Enabling Multicast-Aware Bridging

The PowerHub supports VLANs for IP routing. A VLAN is an IP interface configured on multiple segments. When the PowerHub receives a packet on an IP Multicast virtual interface that maps to multiple physical segments, it can bridge the packet to other segments and simultaneously route it to other virtual interfaces, transmitting the same copy of the packet on all segments. When this occurs, the time-to-live (TTL) of the bridged packets, as well as the routed packets, is reduced by one. Because this procedure avoids copying the packet again, it results in improved performance. Because most multicast applications use a large TTL value, reducing by one hop when bridging occurs should not significantly affect performance.

If it is not desired that IP Multicasting make its forwarding decisions upon the receipt of membership reports on a port, Multicast bridging can be disabled by issuing the following command:

```
enable multicast-aware-bridging
```

To disable Multicast bridging, issue the following command:

```
disable multicast-aware-bridging
```

The default is **`disabled`**.

CHAPTER 17 Configuring IP/RIP

In a routed environment, routers communicate with each other to keep track of available routes. The PowerHub routing software implements standard Routing Information Protocol (RIP) for exchanging TCP/IP route information with other routers. RIP uses User Datagram Protocol (UDP), an industry-standard connectionless protocol, for sending and receiving packets between the PowerHub and other devices.

This chapter describes how to use `ip/rip` subsystem commands to perform the following tasks:

- Display the RIP configuration
- Configure RIP parameters for IP networks
- Display and clear RIP statistics
- Enable RIP Bridging (used only when IP traffic crosses the PowerHub on a VLAN)

Configuring IP/RIP

17.1 Accessing the RIP Subsystem

To access the `ip/rip` subsystem, issue the following command at the runtime command prompt:

```
ip/rip
```

17.2 Displaying the RIP Configuration

The **config show** command is used to display the current RIP configuration for a specified IP interface address. The results from this command is:

```
57:PowerHub:ip/rip# config show
RIP Configuration
-----------------
RIP Bridging: enabled
                          Rpt    Rpt  Acpt  Auth   Key
I/F Addr TX  RX  Poison  Static  Def  Def   Type   ID   Txtype  Rxtype
-------- --- --- ------  ------  ---  ----  ----   ---  ------  ------
19.0.0.1 yes yes  yes     yes    yes   yes  none   ---   rip2   both
```

17.2.1 Configuring RIP Parameters

The **`rip nenable`** command can be used to set one or more requested RIP parameters for an IP interface address and add the information to the RIP update control table. The syntax for this command is:

`rip nenable` *<params> <ipaddr>*

<params> Specifies either a comma-separated list of parameters or **`all`** for all parameters. Parameters are:

talk|ta Specifies to add an entries to the RIP packets for the specified subnet.

listen|li Specifies to listen for RIP packets received on the specified IP interface.

poison|po When a learned route from the specified IP interface goes down, specifies one of the following actions:

If **no** is specified, stop reporting the route.

If **yes** is specified, report the route one more time, but with a metric (hop count) of 16, which is infinity as far as RIP is concerned. Other routers learn immediately that the route is down.

RptStaticRt|rs Specifies whether static routes for the specified IP interface address are reported in RIP packets sent out the segment containing the interface.

RptDefaultRt|rd When RIP packets are generated on the interface, specifies whether to report the default route, if any, in its route table.

AccptDefaultRt|ad Specifies whether default routes for the specified IP interface address are reported to the RIP update control table.

NOTE If the *<params>* is not entered, all parameters except **`poison`** are set to **`yes`**.

Following is an example of the use of the **`rip nenable`** command:

```
56:PowerHub:ip/rip# rip nenable ta,li,po,rd,rs 19.0.0.1
Okay
57:PowerHub:ip/rip#
```

If RIP parameters are already set and it is decided to change the RIP control type from per-VLAN to per-segment, a warning message is displayed before allowing the change. The change of control type automatically clears the per-VLAN RIP parameters from all pertinent tables. The new parameters are supplied from one of two sources:

- If the configuration is saved in a `config` file, the RIP parameters are whatever was saved in that file.
- If the configuration is not saved in a `config` file, the RIP parameters are supplied by switch defaults.

This applies to all IP interfaces defined before you changed the RIP control type.

If a VLAN is deleted, the parameters associated with it are removed from all associated tables. If a segment is added to a VLAN, the new segment adopts the existing RIP parameters. If a segment is deleted from a VLAN, the RIP parameters remain in effect for those segments still assigned to the VLAN.

The **`rip-bridging enable`** command can be used to set one or more requested RIP parameters for an IP interface address and add the information to the RIP update control table. The syntax for this command is:

```
rip-bridging|rb enable|disable
```

Following is an example of the use of the **`rip-bridging enable`** command:

```
56:PowerHub:ip/rip# rb enable
Okay
57:PowerHub:ip/rip#
```

17.2.2 Enabling Acceptance of Default Routes

RIP is used to convey information about routes to destinations, which may be individual hosts, networks, or a special destination used to convey a default route. To enable acceptance of a default route in updates sent to the network, issue the following command:

`ad nenable` *<ifaddr>*

<ifaddr> Specifies the IP interface address for which acceptance of a default route is enabled. Specify a specific IP interface address or a comma-separated list of addresses.

To disable the acceptance of default routes, issue the following command:

ad ndisable *<ifaddr>*

<ifaddr> Specifies the IP interface address for which to disable acceptance of a default route. Specify a specific IP interface address or a comma-separated list of addresses.

17.2.3 Enabling Authentication of RIP Updates

To enable authentication of RIP Version 2 updates sent to the network, issue the following command:

auth nenable *<ifaddr>*

<ifaddr> Specifies the IP interface address for which authentication of RIP Version 2 updates is to be enabled. Specify a specific IP interface address or a comma-separated list of addresses.

To disable authentication of RIP Version 2 updates sent to the network, issue the following command:

auth ndisable *<ifaddr>*

<ifaddr> Specifies the IP interface address for which authentication of RIP Version 2 updates is to be deleted. Specify a specific IP interface address or a comma-separated list of addresses.

17.2.4 Setting the Authorization String on a VLAN

To set an authorization sting or a key identifier on a specified VLAN, issue the following command:

auth nset [-k *<keyid>***|***<password>***]** *<ifaddr>*

-k<keyid> Specifies the value to be used as the Authentication Key that has a simple password value. If a string shorter than 16 octets is supplied, the string is left-justified and padded to 16 octets, on the right, with nulls (0x00).

<password> Specifies the type of Authentication used on the interface.

<ifaddr> Specifies the IP interface address for which to set an authorization sting or a key identifier on a specified VLAN. Specify a specific IP interface address or a comma-separated list of addresses.

To unset an authorization sting or a key identifier on a specified VLAN, issue the following command:

auth nunset [-k *<keyid>***|***<password>***]** *<ifaddr>*

<ifaddr> Specifies the IP interface address for which to unset an authorization sting or a key identifier on a specified VLAN. Specify a specific IP interface address or a comma-separated list of addresses.

17.2.5 Enabling Report of Learned Routes

To enable the reporting of learned routes in updates sent to the network, issue the following command:

```
rl nenable <ifaddr>
```

<ifaddr> Specifies the IP interface address on which to enable the advent of routes in updates sent to the network.

To disable the reporting of learned routes in updates sent to the network, issue the following command:

```
rl ndisable <ifaddr>
```

<ifaddr> Specifies the IP interface address on which to enable the advent of routes in updates sent to the network.

17.2.6 Setting the Receive and Transmit Type on a VLAN

To set the receive type on a specified VLAN, issue the following command:

```
rxtype nset rip1|rip2|both <ifaddr>
```

rip1|rip2|both Specifies the receive type of RIP-1 packets, RIP-2 packets, or both RIP-1 packets, RIP-2 packets on a VLAN.

<ifaddr> Specifies the IP interface address for which to set receive type on a specified VLAN. Specify a specific IP interface address or a comma-separated list of addresses.

To set the transmit type on a specified VLAN, issue the following command:

```
txtype nset rip1|rip1c|rip2 <ifaddr>
```

rip1|rip1c|rip2 Specifies the following type of RIP packets to be transmitted on a VLAN:

rip1 Specifies that RIP-1 messages are sent.

rip1c Specifies that RIP-2 messages are sent broadcast.

rip2 Specifies that RIP-2 messages are sent multicast.

<ifaddr> Specifies the IP interface address for which to set transmit type on a specified VLAN. Specify a specific IP interface address or a comma-separated list of addresses.

17.3 Displaying and Clearing RIP Statistics

The `rip` subsystem maintains statistics on RIP packets that it transmits and receives. The **`stats`** command is used to display statistics. Statistics accumulated since the last system reset, or since the most recent statistics clear can be displayed. The syntax for the **`stats show`** command is:

`stats [show] [-t]`

-t Displays statistics accumulated since the last switch reset. If this argument is not used, the statistics accumulated since the last statistics clear are displayed.

Following is an example of the information displayed by this command:

```
54:PowerHub:ip/rip# stats

RIP Packet Statistics (count since last stats clear):

RIP processing queue:

                       Free entries:  150

RIP route timeout queue:

                       Free entries:  2048
```

Use the **`stats clear`** command to clear statistics. As soon as this command is issued, the PowerHub clears the counters for statistics collected since the last statistics clear. Statistics accumulated since the last reboot are not cleared.

17.4 Bridging RIP Updates Across VLANs

By default, the PowerHub does not bridge RIP updates, even within an IP VLAN. Instead, the software routes the RIP updates, incrementing the metric on each route in the update by at least 1. (Routes are incremented by more than 1 if an additional cost was manually added to an IP route listed in the IP route table.)

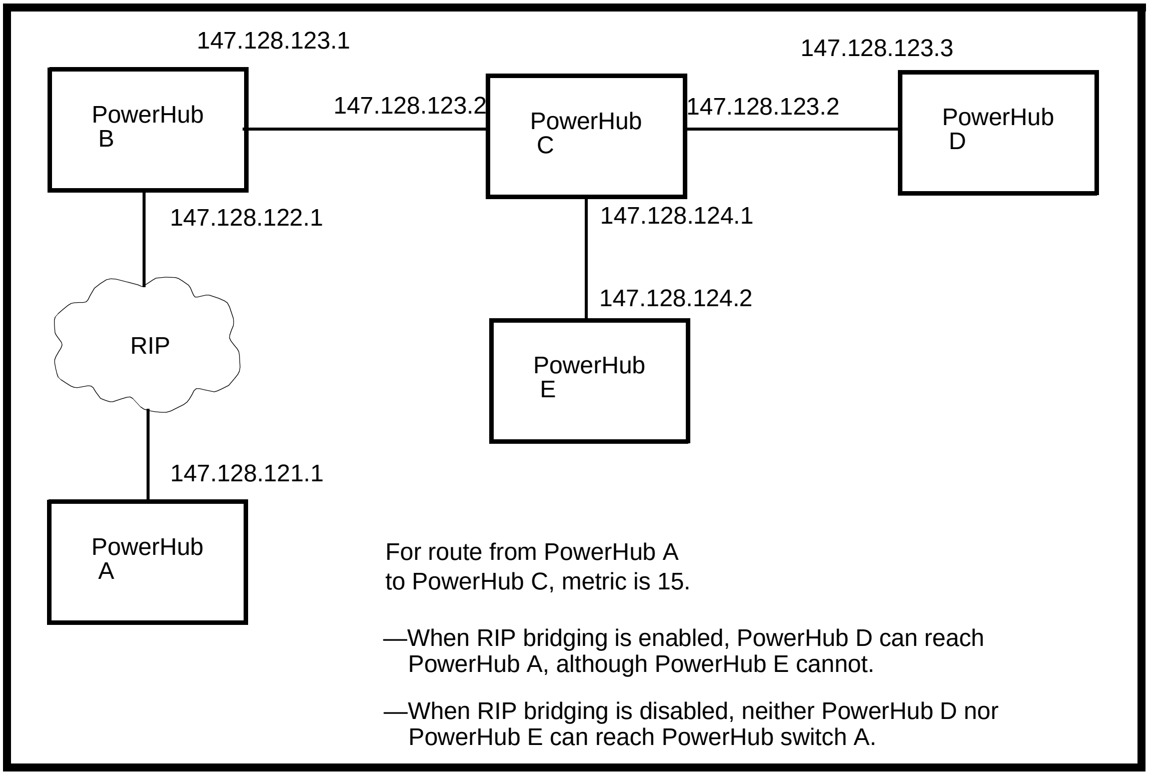
When RIP bridging is enabled, the PowerHub still routes RIP updates when they go from one network to another, and increments the metrics for the routes. However, the software bridges RIP updates when they go from one segment to another within a VLAN. When the software bridges a RIP update, the metrics for the routes contained in the update are not incremented.

Use **`rip-bridging enable`** to configure the PowerHub to bridge RIP updates across IP VLANs. When the software bridges a RIP update, the metric associated with routes in the update is not incremented. Therefore, a hop is not necessarily added to the route. The syntax for this command is:

```
rip-bridging|rb [enable|disable]
```

enable|disable Enables or disables the RIP bridging feature. The default is **`disable`**. If this optional argument is omitted, the current status of the feature is displayed.

Figure 17.1 shows an example of what RIP bridging does.



For route from PowerHub A to PowerHub C, metric is 15.

—When RIP bridging is enabled, PowerHub D can reach PowerHub A, although PowerHub E cannot.

—When RIP bridging is disabled, neither PowerHub D nor PowerHub E can reach PowerHub switch A.

Figure 17.1 - Example of RIP Bridging

In the above figure, two segments on PowerHub C have been configured with IP address 147.128.123.2. Because both segments use the same IP address, they are in a VLAN. IP traffic that normally is routed between different IP networks is bridged within the VLAN.

Suppose that the cost of the route from PowerHub C to PowerHub A is 15 hops. Because 15 hops is the maximum number of hops allowed by RIP, an additional hop would make PowerHub A unreachable. In the network shown above, if RIP bridging is disabled, PowerHub D and E cannot reach PowerHub A. If RIP bridging is enabled, PowerHub D can reach PowerHub A because PowerHub C does not increment the metric for the route to A before reporting the route to PowerHub D.

CHAPTER 18 Configuring IP/OSPF

This chapter lists the PowerHub requirements for using Open Shortest Path First (OSPF) and describes basic features of OSPF. For complete information about OSPF, refer to RFC 2178. The PowerHub implementation of OSPF is based on this RFC.

18.1 Accessing the IP/OSPF Subsystem

To access the ip/ospf subsystem, issue the following command at the runtime prompt:

```
ip/ospf
```

18.2 Configuring a PowerHub Switch as an OSPF Router

The PowerHub can be configured as the following types of OSPF router:

- Internal
- Backbone[1]
- Area Border
- Autonomous System Border

An OSPF router can function as more than one of the router types listed above. For example, a PowerHub that has interfaces attached to the backbone and to other OSPF areas can function both as a Backbone router and as an Area Border router.

Generally, it is not necessary to worry about the differences among these router types. The OSPF software determines how the PowerHub is being used based upon the network configuration.

[1] No special configuration steps are required to use the PowerHub as a Backbone router. If the PowerHub is configured as an Area Border router or an Autonomous System Border router places the switch on the backbone, the PowerHub also functions as a Backbone router.

Unless OSPF areas are configured using the area add command, the PowerHub assumes that the PowerHub is configured as a Backbone router. In addition, the software automatically configures the area ID 0.0.0.0 for the backbone.

To configure the PowerHub for OSPF routing, perform the following tasks. These tasks apply to all OSPF router types.

- Allocate memory for OSPF.
- Add IP interfaces (if interfaces are not already configured) *Chapter 15, Configuring IP Routing*.
- Enable IP forwarding (if not already enabled) *Chapter 15, Configuring IP Routing*.
- Assign the OSPF router ID.

Depending upon the type of OSPF router to be used, it may be necessary to perform some additional configuration tasks.

- If the PowerHub is to be used as an Interior router or an Area Border router, add OSPF areas, then add OSPF interfaces to the areas.
- If the network contains areas that are not connected to the backbone and are not connected to each other, and the Area Border router for one of these areas is not a PowerHub, it may be necessary to create virtual links.
- If the PowerHub is to be used as an Autonomous System Border router, enable the PowerHub as this type of router.

Finally, after completing the OSPF configuration steps listed above, enable OSPF routing. The following sections describe how to perform these tasks.

18.2.1 Allocating Memory

Aportion of main memory must be allocated for the `ospf` subsystem. It cannot be accessed if memory is not allocated. To allocate memory for the `ospf` subsystem, issue the following command:

```
getmem
```

18.2.2 Assigning the OSPF Router ID

Each OSPF router within the Autonomous System must have a unique OSPF router ID. The OSPF router ID is a 32-bit address in IP format. The software does not assign an address automatically.

Any 32-bit address can be used for the OSPF router ID. However, FORE Systems recommends that one of the IP addresses configured on the PowerHub be used. Using one of the IP addresses on the PowerHub ensures that OSPF IDs remain unique. If an IP address configured on the switch is choosen, this does not affect IP or OSPF. That is, the software does not establish a special relationship between the IP address chosen and the OSPF software.

By requiring that an IP address configured on the switch be used, the PowerHub OSPF software ensures that the OSPF router ID remains unique regardless of changes in the network. To assign the OSPF router ID, issue the following command:

router-id set *<router-id>*

<router-id> Specifies the OSPF router ID in dotted decimal notation (xxx.xxx.xxx.xxx, where each “x” is an integer from 0 through 9).

NOTE Define the OSPF router ID only when OSPF routing is disabled. To verify that OSPF routing is disabled, issue the **config show** command.

Following is an example of this command.

```
2:PowerHub:ip/ospf# router-id set 1.1.1.1
```

18.2.3 Displaying the Router-ID

To display the router-id table, issue the following command:

router-id [show]

Following is an example of this command.

```
378:PowerHub:ip/ospf# router-id show
OSPF Router                          : memory available
OSPF Routing                         : Disabled
OSPF Router ID                       : NOT DEFINED!
OSPF Version number                  : 2
OSPF Autonomous System Boundary Router: Enabled
Automatic Virtual Link Feature       : Enabled
379:PowerHub:ip/ospf# help router-id
```

18.2.4 Enabling OSPF

To enable OSPF, issue the following command:

ospf enable

To disable OSPF, issue the following command:

ospf disable

18.2.5 Enabling the PowerHub as a System Border Router

The PowerHub can be enabled to function as an Autonomous System Border router. A switch enabled to be an Autonomous System Border router automatically exports OSPF routes to the networks outside of the OSPF Autonomous System and imports routes from the networks outside the Autonomous System. To enable or disable the PowerHub as an Autonomous System Border router, issue the following command:

```
asbd enable|disable
```

enable|disable Specifies whether to enable or disable the PowerHub to function as an Autonomous System Border router. If **`enable`** is specified, the PowerHub can exchange route information between RIP and OSPF. If **`disable`** is specified, the software cannot exchange route information. The default is **`disable`**.

To view the changes you've made, issue the following command:

```
asbd [show]
```

Following is an example of this command:

```
4:PowerHub:ip/ospf# asbd show

OSPF Router:                           memory available
OSPF Routing:                          Enabled
OSPF Router ID:                        1.1.1.1
OSPF Version number:                   2
OSPF Autonomous System Boundary Router:Enabled
Automatic Virtual Link Feature:        Enabled
```

18.2.6 Setting the Automatic Virtual-Link Feature

The automatic virtual-link feature builds virtual links between the areas that are not connected to the backbone. By building the virtual links, the PowerHub ensures that complete route information reaches all the OSPF routers in the Autonomous System.

NOTE

> The automatic virtual-link feature establishes virtual links only between PowerHubs. To create a virtual link between the PowerHub and another type of router, use the **`vlink add`** command on the PowerHub. See the documentation for your other router for information on establishing that router's end of the virtual link.

The automatic virtual-link feature is enabled by default. To disable (or re-enable) the feature, issue the following command:

`auto-vlink enable|disable`

enable|disable Specifies whether to enable or disable the automatic virtual-link feature.

18.2.6.1 Displaying the Virtual-Link Table

To display the Virtual-Link table, issue the following command:

`auto-vlink [show]`

Following are the results produced by this command:

```
335:PowerHub:ip/ospf# auto-vlink
OSPF Router                           : memory available
OSPF Routing                          : Disabled
OSPF Router ID                        : NOT DEFINED!
OSPF Version number                   : 2
OSPF Autonomous System Boundary Router: Enabled
Automatic Virtual Link Feature        : Enabled
336:PowerHub:ip/ospf# help auto-vlink
```

18.2.7 Adding an OSPF Interface to an Area

An OSPF interface is automatically added to the PowerHub when an IP interface is added. The OSPF interface has the same address as the IP interface. When OSPF routing is enabled, the interface is automatically added to the backbone area (0.0.0.0). To show the OSPF interface, issue the following command:

`interface|it [show]` ***`<ip-addr>`***

<ip-addr> Specifies the IP address of the interface. Specify the interface in dotted decimal notation (xxx.xxx.xxx.xxx, where each “x” is an integer from 0 – 9).

NOTE Type-of-Service (TOS) cannot be specified. The PowerHub uses TOS 0 (zero, the IP TOS).

18.2.8 Using the NSET Command

In most Autonomous Systems, the PowerHub defaults for the OSPF interface parameters are appropriate for the Autonomous System. However, if change to a specific interface parameter is required, use the following command to do so.

```
nset <ip-addr> [ar <area-id>] [auth <key-str>]
   [cost|c <cost>] [priority|p <priority>]
  [xdelay|x <trans-delay>] [rint|r <rxmt-int>]
  [hint|h <hello-int>] [rdint|d <rtr-dead-int>]
```

<ip-addr> Specifies the IP address of the interface. Specify the interface in dotted decimal notation (xxx.xxx.xxx.xxx, where each “x” is an integer from 0 – 9).

The IP address must already be present in the IP interface table before it can be used to create an OSPF interface.

[ar <area-id>] Specifies the OSPF area in which the OSPF interface is being placed. An OSPF interface can belong to only one area. The area must already be configured (using the **area add** command.

[auth <key-str>] Specifies the authentication string. For a simple password, specify any combination of up to eight numbers, letters, and special characters. For MD5 authentication, specify any combination of up to 16 numbers, letters, and special characters. The authentication string is case-sensitive.

If the area to which this interface is being added does not require an authentication string, use empty quotation marks ("").

[cost|c <cost>]

Specifies the cost of using this interface. The PowerHub advertises the cost in Router Links Advertisements. Specify a cost from **1** through **32**. This parameter does not have a default value. The cost depends upon the wire speed of the segment on which the interface is being added. Unless the cost needs to be changed, FORE Systems recommends that this argument be omitted and use the value determined by the PowerHub.

[priority|p <priority>

Specifies this interface's priority during the election process for the Designated Router (DR). The interface with the highest priority number is elected as the DR. The interface with the second-highest priority number is elected as the Backup Designated Router (BDR).

Specify a priority from 0 through 255. Priority increases from **1** (lowest) to **255** (highest). A priority of **0** (zero) makes this interface ineligible for becoming the DR. The default is **1**.

If all OSPF interfaces within an Autonomous System have the same priority, the DR and BDR are elected based on the interface addresses. The interface with the highest OSPF address is elected as the DR. The interface with the second-highest OSPF address is elected as the BDR.

NOTE

Generally, an OSPF router has only one interface per area. If the PowerHub has multiple interfaces to the same area, the interface priority still applies.

[xdelay|x <transdelay>]

Specifies the interface transmission delay, which is the estimated number of seconds it takes to transmit a Link State Update packet over this interface. The PowerHub adds the transmission delay specified to the ages of the LSAs contained in the Link State Update packets sent on this interface.

Specify a delay from **1** through **3600**. The default is **1**. Refer to RFC 2178 for information about choosing transmission delay.

[rint|r <rxmt-int>] Specifies the retransmission interval. The retransmission interval is the number of seconds between transmissions of LSAs to the OSPF routers adjacent to this interface. The retransmission interval also is used when transmitting Database Description and Link State Request packets.

Specify an interval from **1** through **3600**. The default is **5**.

[hint|h <hello-int>] Specifies the hello interval. The hello interval is the number of seconds between transmission of Hello packets on this interface. Specify an interval from **1** through **65536**. The default is **10**.

NOTE

The hello interval (**hint**) and the router-dead interval (**rdint**) must match on neighbors. That is, the values for these parameters must match the values on the neighbor for these parameters. If the OSPF neighbor also is a PowerHub system, ensure that the values match by accepting the defaults for these parameters. If the neighbor is not a PowerHub, the value on the neighbor or on the PowerHub may need to be changed so that the values on both routers match.

[rdint|d <rtr-dead-int>] Specifies the router-dead interval. The router-dead interval is the number of seconds the OSPF neighbors should wait before declaring that the PowerHub, as an OSPF router, is down.

Specify a router-dead interval from **1** through **65536**. Specify an interval that is an even multiple of the Hello interval. The default is **40**.

18.2.9 Adding an OSPF Area

When OSPF routing is enabled, the PowerHub automatically creates an OSPF area for the network backbone. The area ID for the backbone is always 0.0.0.0. Depending upon how the network is organized, additional OSPF areas may need to be added. To add an OSPF area to the PowerHub, issue the following command:

```
area add<area-id> [<auth-type>] [stub-area-cost|sac <cost>]
```

add Specifies to add an OSPF area.

<area-id> Specifies the area ID. Specify the area ID in dotted decimal notation (xxx.xxx.xxx.xxx, where each “x” is an integer from 0 – 9). The are ID must be unique within the Autonomous System.

NOTE The area ID 0.0.0.0 is reserved for the Autonomous System’s backbone and is already present.

<auth-type> Specifies the authentication type. Specify one of the following:

none|no Specifies that the OSPF area being added does not use authentication.

simple-password|sp Specifies that a password is required for OSPF packets sent within this area.

md5|m Specifies that MD5 authentication is required for OSPF packets sent within this area. See RFC 1321 for information about MD5 authentication.

The PowerHub default is none (no authentication).

When an OSPF interface is added to this area (using the **`interface`** command), specify the actual simple password or MD5 authentication key ID.

NOTE

All OSPF routers in an area must have the same authentication type and the same authentication string. Also, all OSPF routers on a particular network should use the same authentication string.

<stub-area-cost|sac <cost>

Specifies that the area is a stub area. Configuring an area as a stub area reduces OSPF overhead in the network by reducing the amount of OSPF route information flooded to the OSPF routers in the stub area.

The OSPF software does not flood external routing information (information about other Autonomous Systems) into the stub area. Internal routers in the stub area reach Autonomous Systems by using the default route to the stub area's Area Border router.

The OSPF software advertises the default route automatically. Note that a stub area's default route is unrelated to the default routes you can define in the `ip` subsystem. OSPF uses the default routes it defines in preference to manually configured default routes.

The cost is the metric for the default route out of the stub area. The stub area's Area Border router advertises the cost as part of the default route. You can specify a value from **`1`** through **`65535`**. The default is **`1`**.

18.2.9.1 Deleting an OSPF Area

To delete an OSPF area, issue the following command:

```
area delete|del <area-id>|all
```

delete|del

Deletes an OSPF area from the PowerHub.

<area-id>|all

Specifies the area to delete. To delete all OSPF areas defined on this PowerHub, specify **`all`**.

NOTE

Disable OSPF routing before deleting an area. The backbone area (0.0.0.0) cannot be deleted.

18.2.10 Displaying an OSPF Area

The **area|ar [show|sh] [*<area-id>*]** command is used to display information about the OSPF areas configured on the PowerHub.

Following are some examples of the information displayed by this command. In the following example, information is displayed for all the OSPF areas configured on the PowerHub.

```
12:PowerHub:ip/ospf# area show
            Auth  ImportAS  Numberof  #Area  #AS  Number of  Stub Area
Area Id     Type  ExtLSAs   SpfRuns   Bdr    Bdr  Area LSAs  Cost
----------- ----  --------  --------  -----  ---  ---------  --------
0.0.0.0      no   Enabled   12        4      4    13         -----
1.1.1.1      no   Enabled   12        2      2    15         -----
1.2.3.4      md5  Enabled   12        0      0    0          -----
2.3.4.5      sp   Enabled   12        0      0    0          -----
3.3.3.3      no   Enabled   12        1      1    16         -----
33.0.33.0    no   Enabled   12        0      0    0          -----
33.33.33.33  no   Enabled   12        0      0    0          -----
```

In the following example, information is displayed for a specific OSPF area.

```
12:PowerHub:ip/ospf# area show 1.2.3.4
            Auth  ImportAS  Numberof   #Area  #AS  Number of  Stub Area
Area Id     Type  ExtLSAs   SpfRuns    Bdr    Bdr  Area LSAs  Cost
----------- ----  --------  --------   -----  ---  ---------  ---------
1.2.3.4     md5   Enabled    12         0     0     0         ----
```

The fields in this display show the following information:

Area ID Displays the OSPF area ID assigned using the **area add** command. The area ID is a 32-bit integer expressed in dotted decimal notation. The area ID 0.0.0.0 is the backbone area ID and is added automatically by the PowerHub.

Auth Type Displays the authentication type assigned for this area using the **area add** command. The authentication type can have one of the following values:

no No authentication is required for this area.

sp A simple password is required for this area.

md5 MD5 authentication is required in this area. See RFC 1321 for information about MD5.

Import AS Ext LSAs Specifies whether this area is configured to import external LSAs from other Autonomous Systems. The value can be Enabled or Disabled. To change the state of this parameter, use the **`asbd enable|disable`** command.

Number of SPF Runs Indicates the number of times the PowerHub has calculated this area's intra-area route table. This number is reset to zero if OSPF routing is disabled, reboot the software, or power down the PowerHub.

# Area Bdr Indicates the number of Area Border routers that can be reached from this area.

# AS Bdr Indicates the number of Autonomous System Border routers that can be reached from this area.

Number of Area LSAs Indicates the number of LSAs in this area's LSA database. This number does not include external LSAs.

Stub Area Cost If this area is a stub area, the metric for the stub area is indicated in this field. If this area is not a stub area, this field contains dashes (-----). A stub area's metric can be assigned when adding the area using the **`area add`** command.

18.2.11 Adding Network Ranges

It is not necessary to add network ranges to OSPF areas. The PowerHub automatically advertises all the networks on all the OSPF interfaces on the switch to other OSPF routers. Network ranges can be added to reduce OSPF overhead or to hide certain networks from other OSPF routers.

When a network range is added to an area, link-state information for the networks within the range is summarized in the LSAs sent by the switch to its OSPF neighbors. Therefore, if there are many networks within an area, adding the networks as a network range can help reduce OSPF overhead.

In addition, the **noadv** argument can be used with the **`net-range`** command to prevent the switch from advertising routes to the networks within a network range. When the switch sends LSAs to its neighbors, LSAs for the networks in the hidden network range are not sent to the switch's neighbors. Therefore, other routers in the Autonomous System do not learn about the hidden networks.

NOTE

None of the networks within the network range added to an area can be in other areas.

To add a network range to an OSPF area, issue the following command:

```
net-range add <area-id> <net> <mask> [noadv|na]
```

<area-id> Specifies the OSPF area. The area must already be added to the switch. To add an area, use the **area add** command.

<net> Specifies an IP network address in dotted decimal notation (xxx.xxx.xxx.xxx, where each “x” is an integer from 0 – 9). The address specified is ANDed with the subnet mask specified for the *<mask>* argument.

<mask> Specifies the IP mask associated with the IP network address specified for the <*net*> argument. The mask indicates the portion of the IP network address that is to be regarded as the network portion of the address. Specify the mask in dotted decimal notation (ex: 255.255.255.0).

noadv|na Prohibits the OSPF software from advertising this network range in the LSAs transmitted by the switch to its OSPF neighbors. If this argument is used, other OSPF routers do not learn about the presence of the network range.

In the following example, the network range specified by IP address 200.200.200.0 and subnet mask 255.255.255.0 is added to area 1.1.1.1. When area 1.1.1.1 sends LSAs to other areas, the LSAs contains summary information for the networks within the network range, instead of detailed link-state information for each network within the network range.

```
9:PowerHub:ip/ospf# net-range add 1.1.1.1 200.200.200.0 255.255.255.0
OSPF: Net "200.200.200.0" with Mask "255.255.255.0" added to area "1.1.1.1"
```

If the **noadv** argument had been specified with the command, the area would not report the networks within the specified network range.

18.2.12 Deleting Network Ranges

To delete a network range, issue the following command:

`net-range delete|del` ***`<area-id> <net> <mask>`***

<area-id> Specifies the OSPF area.

<net> Specifies the IP network address.

<mask> Specifies the subnet mask associated with the IP address.

Here is an example of this command.

```
10:PowerHub:ospf# net-range del 1.1.1.1 200.200.200.0 255.255.255.0
OSPF: Net "200.200.200.0" with Mask "255.255.255.0" deleted from area "1.1.1.1"
```

After a network rangehas bene deleted, the PowerHub sends detailed link-state information for each network, instead of summarizing the link-state information for the entire range.

18.2.13 Displaying Network Ranges

Use the **`net-range show [`***<area-id>***`]`** command to display information about the network ranges assigned to the areas configured on the PowerHub. If the optional <*area-id*> argument is omitted, summary information is displayed for all the network ranges in all the areas. To display network-range information for a specific area, use the <*area-id*> argument.

Here is an example of the information displayed by the **`net-range show`** command. In this example, the optional<*area-id*> argument is omitted. Only one network range is listed in the display, indicating that only one OSPF network range has been configured.

```
19:PowerHub:ospf# net-range show
Area ID            Net                Mask               Advertise
---------------    ---------------    ---------------    ---------
1.1.1.1            200.200.200.0      255.255.255.0      Enabled
```

The fields in this display show the following information:

Area ID The OSPF area that contains the network range.

Net The IP address of the network or subnet portion of the network range. The network number is ANDed with the subnet mask (see the Mask field) to make the network range.

Mask The subnet mask that is ANDed with the network number (see the Net field) to make the network range.

Advertise Indicates whether this network range is advertised to other areas. The advertise state can be Enabled or Disabled. The advertise state is enabled by default. To prevent from advertising the network range to other areas, use the **noadv** argument with the **net-range** command.

18.2.14 Displaying OSPF Neighbors

The **neighbor show** command is used to display information about OSPF neighbors. Here is an example of the information displayed by this command.

```
18:PowerHub:ip/ospf# neighbor show
IP Address     Router ID     Pri   State     Events    RTrQ
-------------- ----------    ---   ------    ------    -----
129.213.72.2   5.5.5.5       1      full       6        0
150.1.100.3    3.3.3.3       1      full       6        0
```

The fields in this display show the following information:

IP Address Neighbor's interface IP address.

Router ID The ID of the OSPF router that contains the neighbor.

Pri The priority of the OSPF router that contains the neighboring interface. The priority is used when the PowerHub elects a DR and a BDR. If the priority is 0 (zero), the OSPF router is ineligible to become the DR or BDR.

State The state of the relationship with the neighboring interface's router. The state can be one of the following:

down The switch has not received recent information from the neighbor.

attempt The switch has not received recent information from the neighbor, but the software is attempting to contact the neighbor by sending Hello packets. The Hello interval can be changed using the **hint** argument of the **nset** command.

init The switch recently received a Hello packet from the neighbor.

two Way Communication between the switch and the neighbor now is bi-directional.

ex start The switch and its neighbor are beginning to exchange their link-state databases.

exchange The switch is sending its link-state database to the neighbor.

loading The switch is sending Link State Request packets to the neighbor, requesting the LSAs that are more recent than the information contained in the link-state database the switch sent to that neighbor. The switch updates its link-state database with the new LSAs received from the neighbor.

full The switch and the neighbor have finished exchanging their link-state databases.

For more information about these states, refer to RFC 2178.

Events The number of times the state of the neighbor relationship (see the State field) has changed. Refer to RFC 2178.

RTrQ The current length of the retransmission queue.

18.2.15 Displaying OSPF Link-State Advertisements

Use the following command to display information about a link-state database:

```
lsdb show [<lsdbid> <rid> <type> <aid>]
```

<lsdbid> Specifies the ID of a specific LSA.

<rid> Specifies the OSPF router ID of the router from which the link-state database was received.

<type> Specifies the LSA type, which can be one of the following types:

r : Router LSA

n : Network LSA

s : Summary LSA

a : Autonomous System Summary LSA

e : External LSA

<aid> Specifies the ID of the area to which the LSA applies.

If the optional arguments are omitted, summary information is displayed for all the LSAs present in the LSA database. To display detailed information about a specific LSA, use the optional arguments.

Here are some examples of the information displayed by this command. In the first example, summary information for all LSAs in the switch's LSA database is displayed.

```
16:PowerHub:ip/ospf# lsdb show
Area Id   Lsdb Type      Link State ID   Router ID   Sequence
--------- -------------  --------------  ----------  -----------
0.0.0.0   routerLink     1.1.1.1         1.1.1.1     -2147483552
0.0.0.0   routerLink     2.2.2.2         2.2.2.2     -2147483303
0.0.0.0   routerLink     3.3.3.3         3.3.3.3     -2147483615
0.0.0.0   routerLink     5.5.5.5         5.5.5.5     -2147483576
0.0.0.0   networkLink    80.100.1.3      3.3.3.3     -2147483635
0.0.0.0   networkLink    129.213.72.2    5.5.5.5     -2147483635
0.0.0.0   summaryLink    87.0.0.0        2.2.2.2     -2147483348
0.0.0.0   summaryLink    150.1.100.0     1.1.1.1     -2147483578
0.0.0.0   summaryLink    150.1.100.0     3.3.3.3     -2147483632
1.1.1.1   routerLink     3.3.3.3         3.3.3.3     -2147483635
1.1.1.1   networkLink    150.1.100.3     3.3.3.3     -2147483646
1.1.1.1   summaryLink    44.0.0.0        3.3.3.3     -2147483640
1.1.1.1   summaryLink    80.100.0.0      3.3.3.3     -2147483640
1.1.1.1   summaryLink    80.200.0.0      3.3.3.3     -2147483640
<example truncated for brevity>
```

The fields in this display show the following information:

Area ID The OSPF area from which the LSA was received.

Lsdb Type The type of LSA.

Link State ID The ID of the LSA, in dotted-decimal notation. The LSA ID is determined by the type of the LSA, as described in Table 18.1:

Table 18.1 - LSA Type to LSA ID

LSA Type	LSA ID
An Internal router's LSA (routerLink).	The originating router's OSPF router ID.
A network LSA (networkLink).	The IP interface address of the network's DR (Designated Router).
A summary LSA (summaryLink).	The destination network's IP address.
An Autonomous System Border router's LSA (asSummaryLink).	The OSPF router ID of the Autonomous System Boundary router described by the LSA.
An Autonomous System Border router's external LSA (asExternalLink).	The destination network's IP address.

Route ID The OSPF router from which the LSA was received.

Sequence The sequence number of the LSA. The sequence number is a 32-bit signed integer. A higher sequence number indicates a more recent LSA. Use the LSA sequence numbers to detect old or duplicate LSAs.

In the following example, detailed information is displayed about a specific LSA.

```
17:PowerHub:ip/ospf# lsdb show 1.1.1.1 1.1.1.1 r 0.0.0.0
Detailed View
Area ID                      : 0.0.0.0
Link State Database Type     : routerLink
Link State ID                : 1.1.1.1
Originating Router ID        : 1.1.1.1
Sequence Number              : -2147483552
Advertisement Age            : 1503
Advertisement Checksum       : ccac
The OSPF Link State Database Advertisement: (26 per line)
00 00 02 01 01 01 01 01 01 01 01 01 80 00 00 60 cc ac 00 30 03 00 00 02 81 d5
48 02 81 d5 48 01 02 00 00 0a 03 03 03 03 96 01 64 01 04 00 00 0a
```

The fields in this display show the following information:

Area ID The OSPF area from which the LSA was received.

Link State Database Type The type of LSA. The LSA can be one of the following types:

`routerLink`	Internal router LSA
`networkLink`	Network LSA \
`summaryLink`	Summary LSA
`asSummaryLink`	Autonomous System Border router LSA
`asExternalLink`	External LSA

Link State ID Area ID The ID of the LSA. The LSA ID depends upon the type of the LSA as defined in Table 18.1.

Originating Router ID Area ID The OSPF router from which the LSA was received.

Sequence Number Area The sequence number of the LSA. The sequence number is a 32-bit signed integer. A higher sequence number indicates a more recent LSA. Use the LSA sequence numbers to detect old or duplicate LSAs.

Advertisement Age Area ID The age, in seconds, of the LSA.

Advertisement Checksum Area The checksum for the LSA.

The OSPF Link State Database Advertisement Area The contents of the LSA, in hexadecimal.

18.2.16 Enabling the Return-Code Prompt

The return-code prompt is intended primarily for automated interactions with the PowerHub command-line interface. To enable printing of command return codes in the next UI prompt, issue the following command:

```
rcprompt enable
```

To disable the return-code prompt, issue the following command:

```
rcprompt disable
```

18.2.17 Adding a Virtual-Link

Depending upon how the OSPF network is configured, it is possible for some areas to be completely disconnected from one another. Areas become disconnected from one another when they are not attached to the backbone and do not share a Border router.

The PowerHub can automatically link disconnected areas using the automatic virtual-link feature. This feature links together PowerHubs configured as OSPF routers when they are separated from one another.

If some of the OSPF routers in your Autonomous System are not PowerHubs, areas that are separated can be linked by defining a virtual link between the areas. The virtual link makes the disconnected areas virtual neighbors. LSAs from an area reach that area’s virtual neighbor by travelling through a transit area. The transit area is an area between the two virtual neighbors that passes traffic between the neighbors.

NOTE The transit area must be added to the OSPF network before configuring the virtual link.

To add a virtual link, use the following command:

```
virtual-link|vlink add <aid> <router-id> [auth <key-str>]
        [xdelay|x <trans-dly>] [rint|r <rxmt-int>]
        [hint|h <hello-int>] [rdint|d <rtr-dead>]
```

The values and defaults for these arguments are the same as the arguments and defaults for the **nset** command.

18.2.18 Deleting a Virtual-Link

To delete a virtual link, issue the following command:

```
virtual-link|vlink delete|del <aid> <router-id>
```

<aid> Specifies the area ID of the transit area. Specify the area ID in dotted decimal notation (xxx.xxx.xxx.xxx, where each “x” is an integer from 0 – 9).

<router-id> Specifies the OSPF Router ID of the virtual neighbor. Specify the router ID in dotted decimal notation (xxx.xxx.xxx.xxx, where each “x” is an integer from 0 – 9).

NOTE Use the **virtual-link del** command to delete a virtual link created by the software automatically using the automatic virtual-link feature. However, if the automatic virtual-link feature is enabled, the software adds the link again. To prevent the software from adding a virtual link again, disable the automatic virtual-link feature by issuing the **auto-vlink disable** command.

18.2.19 Displaying Virtual-Links

Use the following command to display information about a virtual link:

virtual-link|vlink [show] [*<aid> <router-id>*]

If the optional arguments are omitted, summary information is displayed for all the virtual links that exist between this PowerHub and other OSPF routers. To display detailed information about a virtual link, use the optional arguments.

Following is an example of the information displayed by this command. In this example, summary information is displayed. The switch in this example has only one virtual link to another OSPF router.

```
20:PowerHub:ip/ospf# virtual-link show
Area ID      Router ID     IP Address    If State    Nbr State
------------ ------------- ------------- --------    ---------
1.1.1.1      3.3.3.3       150.1.100.3   up          full
```

The fields in this display show the following information:

Area ID The OSPF area on the local side of the virtual link.

Router ID The router ID of the OSPF router on the local end of the virtual link. (The PowerHub OSPF router ID.)

IP Address The IP address of the router on the remote end of the Virtual Link. Routers can have many IP addresses. This IP address is the one assigned to the remote router's segment that connects the remote router to the PowerHub.

IF State The state of the virtual interface. The state can be one of the following:

`up` The interface can be used to send and receive OSPF route information.

`down` The interface is unavailable for sending or receiving OSPF traffic. The interface's link state is be reported as down in LSAs sent from this OSPF router.

Nbr State The state of the virtual interface. The state can be one of the following:

`down` The switch has not received recent information from the neighbor.

`attempt` The switch has not received recent information from the neighbor, but the software is attempting to contact the neighbor by sending Hello packets. The Hello interval can be changed by using the **`hint`** argument of the **`nset`** command.

`init` The switch recently received a Hello packet from the neighbor.

`two Way` Communication between the switch and the neighbor now is bi-directional.

`ex start` The switch and its neighbor are beginning to exchange their link-state databases.

`exchange` The switch is sending its link-state database to the neighbor.

`loading` The switch is sending Link State Request packets to the neighbor, requesting the LSAs that are more recent than the information contained in the link-state database the switch sent to that neighbor. The switch updates its link-state database with the new LSAs received from the neighbor.

`full` The switch and the neighbor have finished exchanging their link-state databases.

In the following example, detailed information is displayed for a specific virtual link.

```
21:PowerHub:ip/ospf# virtual-link show 1.1.1.1 3.3.3.3
Area ID                            : 1.1.1.1
Router ID                          : 3.3.3.3
IP Address                         : 150.1.100.3
Transit Delay                      : 1
Retransmission Interval            : 5
Hello Interval                     : 10
Router Dead Interval               : 60
Authorization Key String           :
Authorization Failures             : 0
Virtual Interface State            : up
Virtual Interface Events           : 1
Virtual Neighbor State             : full
Virtual Neighbor Events            : 5
Virtual Neighbor Retransmission Que : 0
```

The fields in this display show the following information:

Area ID The OSPF area on the local side of the virtual link.

Router ID The router ID of the OSPF router on the local end of the virtual link. (The PowerHub OSPF router ID.)

IP Address The IP address of the router on the remote end of the Virtual Link. Routers can have many IP addresses. This IP address is the one assigned to the remote router's segment that connects the remote router to the PowerHub.

Transit Delay The interface transmission delay for this interface.

Retransmission Interval The retransmission interval for this interface.

Hello Interval The Hello interval for this interface.

Router Dead Interval The Hello interval for this interface.

Authorization Key String The authorization string for the interface. The authorization string is specified by the <*key-str*> argument of the **`interface`** command. If this field is blank, then no authorization string is required for this interface.

Authorization Failures The number of times another OSPF router tried to use this interface but did not supply the correct authorization string.

Virtual Interface State The state of the virtual interface. The state can be one of the following:

`up` The interface can be used to send and receive OSPF route information.

`down` The interface is unavailable for sending or receiving OSPF route information. The interface's link state is reported as down in LSAs sent from this OSPF router.

Virtual Interface Events The number of times the state (see the Virtual Interface State field) has changed since OSPF routing was enabled.

Virtual Neighbor State The state of the relationship with the OSPF router on the remote end of the virtual link. The state can be one of the following:

`down` The switch has not received recent information from the neighbor.

`attempt` The switch has not received recent information from the neighbor, but the software is attempting to contact the neighbor by sending Hello packets. The Hello interval can be changed by using the **`hint`** argument of the **`nset`** command.

`init` The switch recently received a Hello packet from the neighbor.

`two Way` Communication between the switch and the neighbor now is bi-directional.

`ex start` The switch and its neighbor are beginning to exchange their link-state databases.

`exchange` The switch is sending its link-state database to the neighbor.

`loading` The switch is sending Link State Request packets to the neighbor, requesting the LSAs that are more recent than the information contained in the link-state database the switch sent to that neighbor. The switch updates its link-state database with the new LSAs received from the neighbor.

`full` The switch and the neighbor have finished exchanging their link-state databases.

Virtual Neighbor Events The number of times the relationship with the remote end of the virtual link has changed since OSPF routing was enabled. The state is displayed in the `Virtual Neighbor State` field.

18.2.20 Timed Commands

In some router implementations, packet processing can affect timer execution. When multiple routers are attached to a single network, all doing broadcasts, this can lead to the synchronization of routing packets (which should be avoided). If timers cannot be implemented to avoid drift, small random amounts should be added to/subtracted from the timer interval at each firing.

18.2.21 Statistics Command

As soon as OSPF forwarding is enabled, the PowerHub begins collecting OSPF statistics. The **`stats show`** command is used to display statistics or the **`stats clear`** command to clear statistics.

18.2.22 Displaying OSPF Statistics

To display the OSPF statistics, issue the following command:

```
stats show
```

Here is an example of the information displayed by the **stats** command.

```
22:PowerHub:ip/ospf# stats show
External Link-State Advertisements          : 0
Checksum of the External LSA Database       : 0
New Link-State Advertisements originated    : 105
Link-State Advertisements received          : 121
Neighbor Allocation Fails                   : 0
Link-State Advertisement Allocation Fails   : 121
Link-State Database Allocation Fails        : 121
Database Request Allocation Fails           : 0
Retransmission Allocation Fails             : 0
Acknowledge Allocation Fails                : 0
OSPF Area Border Router                     : True
Total Authorization Failures                : 0
Memory Usage: 3872 bytes used out of 524288 available
              6 fragments allocated, 7 total
```

18.2.23 Clearing OSPF Statistics

To clear OSPF statistics, issue the following command:

```
stats clear
```

Here is an example of this command.

```
23:PowerHub:ip/ospf# stats clear
OSPF: Statistics Cleared.
```

The PowerHub clears the counters for the statistics and begins collecting statistics again. Statistics also are cleared if OSPF routing is disabled, the software is rebooted, or the PowerHub is powered down.

CHAPTER 19 Configuring AppleTalk Routing

The PowerHub atalk (AppleTalk) subsystem contains a complete set of AppleTalk Phase-2 commands use with AppleTalk networks and internets. The PowerHub can be configured to be used as an AppleTalk internet router to perform AppleTalk routing on any or all of its segments. The PowerHub can also be configured as a local router or a backbone router, or as any combination of these types of routers. This chapter describes the atalk subsystem commands you can use to perform the following tasks:

- Allocate memory for the AppleTalk subsystem.
- Enable AppleTalk routing.
- Show the current AppleTalk configuration.
- Add and delete an AppleTalk interface.
- Display the AppleTalk interface table.
- Display the AppleTalk route table.
- Display and clear the AppleTalk route cache.
- Add an and delete an AppleTalk zone.
- Display a list of configured or active (static and learned) AppleTalk zones.
- Set the aging time for entries in the AppleTalk ARP table.
- Display and clear the AppleTalk ARP table.
- Display a table of “named objects.”
- Display statistics for AARP, DDP, or ECHO packets.
- Clear packet statistics.
- Test the connectivity to another router.

19.1 Accessing the AppleTalk Subsystem

To access the atalk subsystem, enter the following command from the runtime command prompt:

```
atalk
```

19.2 Getting Started

To set up the PowerHub for AppleTalk routing, perform these steps:

1. Enable the AppleTalk subsystem:
   - Allocate memory for AppleTalk routing. (See Section 19.2.1.)
   - Enable AppleTalk routing. (See Section 19.2.1.2.)
2. Assign AppleTalk zone names to PowerHub segments. (See Section 19.3.1.1.)
3. Assign AppleTalk network (interface) addresses to PowerHub segments. (See Section 19.4.1.)
4. Save your AppleTalk configuration. (See Section 19.2.1.4.)

19.2.1 Enabling the AppleTalk Subsystem

Before the AppleTalk subsystem can be used, sufficient main memory (DRAM) must be allocated for the PowerHub to run the AppleTalk routing subsystem and enable AppleTalk routing.

19.2.1.1 Allocating Memory

NOTE

FORE Systems recommends that memory for the AppleTalk subsystem be allocated immediately after booting the PowerHub to ensure that the memory requested is available. For more information, refer to the *PowerHub Hardware Reference Manual*.

Memory cannot be de-allocated. To free allocated memory, make sure the configuration file does not contain a **getmem** command, then reboot the software.

To allocate memory for the AppleTalk subsystem, issue the following command:

```
getmem atalk
```

19.2.1.2 Enabling AppleTalk Routing

After allocating memory, enable AppleTalk routing. Use the **enable atalk** command to enable AppleTalk routing:

enable|disable atalk

atalk Indicates that AppleTalk routing on the PowerHub is to be enabled or disabled.

enable|disable Specifies whether AppleTalk routing is being enabled or disabled. The default is **disable**.

Here is an example of this command:

```
4:PowerHub:atalk# enable atalk
AppleTalk Routing: Enabled
```

19.2.1.3 Displaying the Current Configuration

Enter the **config show** command to verify that memory is allocated for the `atalk` subsystem and that AppleTalk routing is enabled. The command also displays the aging time for AppleTalk Address Resolution Protocol (AARP) entries. (See Section 19.5.2.)

```
5:PowerHub:atalk# show config
AppleTalk Router: memory available
AppleTalk Routing: Enabled
AARP Aging Timer: 60 minutes
```

In this example, the display produced by the **show config** command shows the following information:

- Memory has been allocated for the AppleTalk subsystem.
- AppleTalk routing is available and enabled.
- The aging time for learned AARP entries is 60 minutes.

19.2.1.4 Saving Your AppleTalk Configuration

After verifying the AppleTalk configuration, save the configuration using the **savecfg** <*file-name*> command, where <*file-name*> is the configuration file name. When the current configuration is saved, the modifications made to use the AppleTalk subsystem are available next time the PowerHub is booted. For information about this command, refer to the *Hardware Reference Manual*.

19.3 Configuring PowerHub Segments for AppleTalk

Before AppleTalk packets can be routed, assign the appropriate zone names and network addresses to one or more PowerHub network ranges. Use the zone commands and interface commands to configure PowerHub network ranges for use with AppleTalk networks.

19.3.1 The Zone Commands

The PowerHub uses Zone Information Protocol (ZIP) to maintain a zone table that contains zone names associated with PowerHub segments. Use the zone commands to add, display, or delete zone names.

19.3.1.1 Adding a Zone Name

The **zone add** command is used to assign a zone name to a network range. A *zone name* is an alphanumeric string up to 32 characters in length. Different zone names can be assigned to each network range, multiple zone names can be assigned to the same network range, or the same zone name to can be assigned multiple network ranges. Zone names are not required for non-seed segments. Moreover, for non-seed segments, the assigned zone names are not used. Assigned zone names are used for seed segments.

The zone name assigned to a PowerHub network range is used by the segment when it attempts to come up as a seed segment. Unless a conflict occurs over the use of the segment as a seed segment, the zone name becomes active for that segment.

Blank spaces can be used in zone names at the beginning, inside, or at the end of a zone name. To add a zone name that contains a leading or trailing blank(s), use double quotes around the entire zone name, including the blank(s). The syntax for the **zone add** command is:

```
zone|zt add [-d]<netrange> <zone>
```

[-d] Specifies the default zone for this netrange.

<netrange> Optionally specifies a specific range of network addresses.

<zone> Specifies the zone name to assign to the specified netrange. A zone name is a string of 32 characters that are not case sensitive. (For example, the zone names ADMINISTRATION and administration are regarded by AppleTalk as identical.

In the example that follows, the **zone add** command is used to assign the AppleTalk zone name Accounting to netrange 113-119.

```
8:PowerHub:atalk# zone add 113-119 Accounting
Okay
```

Here is an example of how to add a zone name that contains a leading blank. In this example, the zone name also contains an internal blank.

```
1:PowerHub:atalk# zone add 120 “Tony”
Okay
```

When AppleTalk zone names are displayed on the PowerHub, the names that contain leading or trailing blanks are displayed with quotation marks to show the locations of the blanks.

Here is an example of how zone names that contain blanks are displayed.

```
2:PowerHub:atalk# zone show -c
AppleTalk Zones Available for Configuration

    Net-Range          Zones
    ---------          ---------------------
    120-120            tony
    113-119            techsupport
    101-112            *is
3:PowerHub:atalk# name-table
Object NameObject TypeZone
POWERHUBRouter  tony
```

When the zone name is displayed in the Chooser on a Macintosh, the blank spaces appear in the zone name but the quotation marks are not displayed. An asterisk (*****) before the zone name indicates that this is the default zone name for this net-range.

19.3.1.2 Displaying the Zone Information

Information for configured zones or for active zones can be displayed. A *configured zone* is a zone created using the **zone add** command. (See Section 19.3.1.) An *active zone* is a zone name that is actively being used on the AppleTalk internet. An active zone can be either a configured zone or a *learned* zone. A *learned zone* is a zone entry learned by the PowerHub from other routers.

19.3.1.2.1 Configured Zones

The **zone show** command is used to display a list of configured zone names assigned to PowerHub segments. The syntax for this command is:

zone|zt show [-c] *<seglist> <zone> <netrange>*

[-c] Specifies configured interface information. Does not specify dynamically entered interface information.

<seglist> Optionally specifies the segments to list the configured zone names.

<zone> Specifies the zone name assigned to the specified netrange.

<netrange> Optionally specifies a specific range of network addresses.

In the example that follows, the **zone show** command is used to display zone names for segments 1.1 and 2.1.

```
11:PowerHub:atalk# zone show
Net-Range      Segments        Zones
---------      --------        -----
 120              1.1          Test_zone
 110              2.1          Test_zone
```

The **zone show** command is used to display information about active zones (both configured and learned).The **zone show** command shows the network address and the name for each currently active AppleTalk zone that is known to the PowerHub. In addition, the table indicates whether the zone that is active on a particular network is that network's default zone. Note that configured zone names that are not in use are not listed.

In the example that follows, the *<zone-name>* argument is used to display only the networks on which the zone name "FORE Systems" is active. The asterisks (**) to the left of the first network address range indicate that the zone name listed under Zone is the default zone name for that network.

```
13:PowerHub:atalk# zone show FORE Systems
    Net          Zone
**  2-2          FORE Systems
    3-3          FORE Systems
```

Each network has one and only one default zone name. However, the same zone name can be used in more than one network, and can be the default zone name in more than one network.

19.3.1.3 Deleting a Configured Zone

The **zone delete** command is used to delete a configured zone name from one or more segments. The syntax for this command is:

zone|zt delete *<net-range> <zone>*

<net-range> Optionally specifies a specific range of network addresses.

<zone> Specifies the segments from which to delete a configured zone. List individual segments, specify a range of segments, or specify **all** for all segments.

When a configured zone name is deleted, the name disappears from the configured zone table. (Use the **`zone show`** command to display this table.)

NOTE

When the **`zone delete`** command is used to remove a configured zone name, the change is immediately apparent in the Configured-Zone table, but does not affect zone names on interfaces that are currently up. The change can affect an interface if that interface is capable of seeding, and the segment on which the interface is defined is brought down, then back up.

Here is an example of the use of the **`zone delete`** command. At command prompt 14, a specific zone name (FORE Systems) is deleted from the net range 120.

```
14:PowerHub:atalk# zone delete 120-120 FORE Systems
15:PowerHub:atalk#
```

19.4 Configuring AppleTalk Interfaces

After zone names are assigned to one or more PowerHub network ranges, network addresses must be assigned to each of these network ranges. Each network address consists of:

- Network address range.
- Combination of <*net*>.<*node*>.[1]
- Optionally, the default zone name.

[1] In some books, this combination of net address and node address is called a "port node address," an "AppleTalk protocol address," or a "DDP address," depending upon the context. This manual and other PowerHub documentation uses the term "network address" to refer to this combination.

19.4.1 Adding a Interface (Network Address)

The **`interface add`** command is used to assign a network address to one or more PowerHub segments. A different network address can be assigned to each net-range, or the same network address can be assigned to multiple net-ranges. When the same network address is assigned to more than one net-range, a VLAN is created. A VLAN is a network that spans two or more net-ranges. A *VLAN* increases the effective bandwidth of an AppleTalk network without creating additional network numbers. The syntax for the **`interface add`** command is:

```
interface|it add [-n] <seglist> <net>.<node> net[range] <x>-<y> [-h]
            <seglist> <net>.<node> net[range]<x>-<y>
```

[-n] Specifies a non-AppleTalk passive backbone.

<seglist> Specifies the segment numbers to assign an AppleTalk network address. Individual segments, a range of segments, or **all** segments can be specified.

NOTE

To configure a segment as a non-seed segment, specify a network address range of 0-0. Do not specify a network address following the address range.

To create multiple non-seeding segments, issue a separate **`interface add`** command for each net. If multiple segments are specified with the same command, a VLAN is created.

To configure a segment for a non-AppleTalk (backbone) net, specify **-n**, rather than an address range. Do not specify a network address. A backbone net connects routers; nodes are not directly attached to the net.

<net>.<node> Specifies the network address assigned to the specified segment. The value specified for *<net>* must be within the range specified by *<start-net>-<end-net>*.

For *<node>* specify a range from 1 through 253.

NOTE

Do not use this argument if configuring a segment as a non-seed segment or for a non-AppleTalk (backbone) net.

Node addresses 254 and 255 are reserved AppleTalk for EtherTalk; do not use these addresses. If use of these addresses is attempted, an error message is displayed.

net[range] Specifies the network range assigned to a specified segment. Specify a range from **1** through **65023**.

<x>-<y> Specifies the network ranges. For example, a network range of 113-119 can be specified.

[-h] Specifies a hard-seed backbone.

Here are some examples of the use of the **interface add** command. In the first example, the network address range 220 through 500 is assigned to segment 2.1. The network address “220.150” indicates the specific AppleTalk node to which segment 2.1 is assigned:

```
19:PowerHub:atalk# interface add 5 0-0
Port 5 Range 0-0 Added
Configured as non-seeding port.
```

The following example shows the command used to configure segment 2.1 as a non-seed segment. (Note that no network address range or network address is specified.)

```
18:PowerHub:atalk# it add 2.1 220.150 net 220-500
Segment 2.1 Range 220-500 DDP Addr 220.150 Added
Configured as non-seeding interface.
```

19.4.2 Displaying Network Address Information

Information about PowerHub segments assigned to an AppleTalk network address can be displayed using the **interface show** command. The syntax for this command is:

interface|it show [-c] ***<seglist> <netrange> <zone>*** **[-a][-z]**

[-c] Specifies configured interface information. Does not specify dynamically entered interface information.

<seglist> Specifies the segment numbers to display AppleTalk network addresses. Individual segments or a range of segments that have AppleTalk interfaces can be specified.

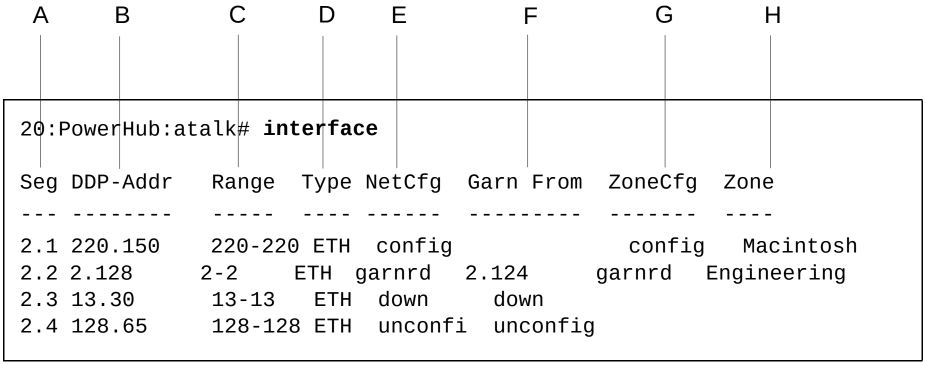
<netrange> Specifies the network range assigned to a specified segment. Specify a range from **1** through **65023**.

<zone> Specifies the zone name t to display network address information.

-a Lists all configured and non-configured segments.

-z Lists all configured and non-configured segments.

Here are some examples of the use of the **`interface` show** command. In the first example, no arguments are used with the command. Network address information is shown for all segments that have AppleTalk interfaces. Only two AppleTalk network addresses are assigned to PowerHub segments. Note that more than one zone can be associated with a segment. In Figure 19.1, three zone names are listed for segment 2.2.



```
20:PowerHub:atalk# interface

Seg DDP-Addr   Range  Type NetCfg  Garn From  ZoneCfg  Zone
--- --------   -----  ---- ------  ---------  -------  ----
2.1 220.150    220-220 ETH  config              config   Macintosh
2.2 2.128     2-2    ETH  garnrd   2.124     garnrd  Engineering
2.3 13.30      13-13   ETH  down     down
2.4 128.65     128-128 ETH  unconfi  unconfig
```

Figure 19.1 - `interface` show Command Details

The table displayed by the **`interface` show** command shows the following information:

A The Seg column lists the segment numbers.

B The DDP-Addr column lists the net address for each segment to which a net address has been assigned. In this example, segments 2.1 and 2.2 are assigned AppleTalk net addresses.

C The Range column lists the net address range assigned to each AppleTalk segment.

D The Type column indicates the media type (in this case, “ETH,” or Ethernet).

E The NetCfg column indicates whether the segment was a seed segment (making the PowerHub a seed router) for the network assigned to the segment, or learned the network information from another router in the net.

The NetCfg column indicates one of four states: config, unconfig, garnrd, or down. The initial state is unconfig. If a segment is the seed segment for a network, config soon appears under the NetCfg column. If the segment is not a seed segment, it instead relies upon another router for seed information. In such a case, when the segment has learned the network address from another router, the state of the segment changes from unconfig to garnrd. If the segment is not configured as a seed segment and there is no other router on the network, the state remains unconfig.

If the state remains `unconfig`, the PowerHub is unable to find a seed for the segment. Check the connections joining the segment to the seed router. If the connections are working properly, the problem might be in the seed router itself.

If a segment has been configured but is attached to a router that is not turned on, or if a segment is attached to a working router but the segment has been either disabled or has not been added to a zone, the segment is listed as down.

If the state is `-cfg`, the segment is part of an AppleTalk VLAN and has gone down. The other segments in the VLAN might still be up.

F The Garn From column indicates the seed router from which the PowerHub got its configuration. If the PowerHub is the seed router, the Garn From field is blank.

G The `ZoneCfg` column indicates whether the interface is a seed router for the zone associated with the segment. Possible states are config, unconfig, garnrd, or down. See the descriptions for NetCfg.

H The Zone column lists the active zone(s) for the segment.

In the following example, the **`-z`** argument is used to limit the display to entries for the specified zone name (in this case, FORE Systems):

```
21:PowerHub:atalk# it show -z FORE Systems
Seg   DDP-Addr   Range  Type   NetCfg  GarnFrom  ZoneCfg  Zone
---   --------   -----  ----   ------  --------  -------  ----
2.2    2.128      2-2   ETH    garnr    2.12     garnrd   FORE Sys.
```

NOTE

If the interface table displays zeros under the DDP-Addr and Range columns, or “down” for the NetCfg and ZoneCfg columns, the segment may be down. If the segment is up, check if AppleTalk routing is enabled. See Section 19.2.1.2 for information on enabling AppleTalk routing.

19.4.3 Deleting a Network Address

The **`interface del`** command is used to remove an AppleTalk network address from a PowerHub segment:

`interface|it del[ete] [-a] <`*`seg-list`*`>`

-a Deletes the AppleTalk network address from a segment(s).

NOTE

Unless the **`-a`** argument is used, each segment to which a network is assigned must be specified in order to delete a network assigned to multiple segments.

<seg-list> Specifies the segments from which to delete the assigned AppleTalk network address. List individual segments, or specify a range of segments.

NOTE

If an AppleTalk network address is deleted, or the zone name with which the deleted address was associated is changed or deleted, it is recommended to wait a minimum of 15 minutes following the zone name change before re-adding the address. This time is needed by the devices in the AppleTalk internet to exchange update information about the network address and zone name changes.

Here is an example of the use of the **interface del** command. In this example, the interface table is displayed to show which interfaces are defined, then the unwanted interfaces are deleted.

```
22:PowerHub:atalk# it show

Seg    DDPAddr   NetRang   Ty    NC       GarnFr    ZC       Zones
---    --------  -------   ---   ------   ------    ------   -----
1.1
1.2
1.3
1.4    220.150   220-230   ETH   config   220.15    config   Macintosh
1.5    2.128     2-2       ETH   garnrd   2.12      garnrd   FORE Sys.
1.6    220.150   220-230   ETH   config   220.23    config   Macintosh
1.7    220.150   220-230   ETH   config   220.23    config   Macintosh
1.8    220.150   220-230   ETH   config   220.23    config   Macintosh
1.9
1.10
1.11
1.12

23:PowerHub:atalk#  it del 1.4
Okay
```

In the example, the network address associated with segment 1.4 is deleted. Because the optional **-a** argument is not used, all the segments with which the network address is associated must be specified.

The following example uses the **interface delete** command with the **-a** argument to delete the same network address:

```
24:PowerHub:atalk# interface del -a 1.6
Okay
```

When the **-a** argument is used, the network address is deleted from all segments to which it is assigned. In this example, network address 220.150, associated with segment 1.6, is deleted from segment 1.6 as well as segments 1.4, 1.7, and 1.8.

19.5 Using the AARP Table

The AARP is used to create and maintain a table of translations between MAC-layer node addresses and AppleTalk node addresses. The AARP table enables the PowerHub to look up the MAC-layer address of another device (node, router, and so on) based on the device's AppleTalk address. Entries in the AARP table facilitate transmission of packets from the PowerHub (acting as an AppleTalk router) to the devices for which MAC-layer addresses are listed. These entries are either static or learned:

Static entry An entry that is created when assigned an AppleTalk network address to a PowerHub segment. Each time a network address is assigned to a segment using the **`interface add`** command, the PowerHub automatically makes a corresponding entry in the AARP table. These entries cannot be deleted unless the corresponding network address is deleted.

Learned entry An entry that the software automatically adds to the AARP table when it learns about a node address from another managed PowerHub or other AppleTalk router, or learns of the node address directly from one of its own segments. The PowerHub deletes learned entries when they are inactive for the *AARP aging time.*

For information on the AARP aging time, see Section 19.5.2. Each entry in the AARP table lists:

- DDP address of the node (also known as AppleTalk node address).
- Type of connection the segment has. There are four types of connections:

  Local Indicates a device is directly attached to the segment.

  Router Indicates the route was dynamically learned. Also indicates another AppleTalk router.

  Bcast Indicates the entry in the AARP table is broadcast to all devices in the network. A broadcast packet is denoted by a node address of **255**.

  blank Indicates a learned address, one that is added by the software. Blank entries also indicate that a node, not a router, is attached.

- MAC-layer address.
- Segment to which the node is attached.

19.5.1 Displaying AARP Entries

The **`arp show`** command is used to display the entries in the AARP table. The syntax for this command is:

`arp|at [show] <`*`seglist`*`> <`*`net.node`*`>`

<net.node> Specifies the network address for which to display AARP entries.

Here are some examples of the use of the **`arp show`** command. In the first example, the command is entered without an argument. The table displayed lists all AARP entries, both static entries and learned entries, for this PowerHub.

```
25:PowerHub:atalk# arp show
  ARP TABLE:
DDP Address   Type    Mac Address         TTL  Segment(s)
-----------   -----   -----------------   ---  ----------
2.5           Local   00-00-ef-02-41-50   10    1.2
2.22                  00-00-94-20-5f-82   20    1.2
2.255         BCast   09-00-07-ff-ff-ff   40    1.2
111.1         Local   00-00-ef-02-41-50   10    1.3,1.4
111.22                00-00-94-21-fd-1c   20    1.3
111.56                00-00-94-21-f2-43   20    1.4
111.255       BCast   09-00-07-ff-ff-ff   40    1.3,1.4
5.1           Local   00-00-ef-02-41-50   20    1.5
5.255         BCast   09-00-07-ff-ff-ff   40    1.5
```

A wildcard (*****) can be specified in place of <*net.node*>. In the following example, all DDP addresses with the net-address “2” are displayed.

```
27:PowerHub:atalk# arp show 2.*
  ARP TABLE:
DDP Address Type    Mac Address         TTL  Segment(s)
----------- -----   -----------------   ---  ----------
2.5         Local   00-00-ef-02-41-50   10    1.2
2.22                00-00-94-20-5f-82   20    1.2
2.255       BCast   09-00-07-ff-ff-ff   40    1.2
```

NOTE

If the AARP table is blank, AppleTalk routing might not be enabled. Use the **`config show`** command to verify that routing is enabled.

19.5.2 Setting the AARP Aging Time

The PowerHub can be configured to maintain the AARP table by specifying the amount of time learned entries can remain inactive in the AARP table before being removed by the software. This time limit is the AARP aging interval and is independent of the aging time for routing table entries. The **`arp set age`** command is used to set the number of minutes a learned AARP entry can be inactive before it is deleted from the AARP table. The syntax for this command is:

`arp set age|saa` ***`<time>`***

<time> Specifies the number of minutes that inactive entries remain in the AARP table. The minimum aging time is **3** minutes.

At command prompt in the example that follows, the **`arp set age`** command is used with the <*time*> argument to change the AARP aging time to 30 minutes.

```
28:PowerHub:atalk# arp set age 30
ARP Age changed to 30 minutes.
```

19.5.3 Clearing the AARP Table

The **`arp clear`** command is used to clear all learned entries from the AARP table. Following is an example of the use of this command:

```
30:PowerHub:atalk# arp clear
Okay
```

19.6 Displaying Route Information

Each PowerHub serving as a router in an AppleTalk internet uses Routing Table Maintenance Protocol (RTMP) to maintain a table of information about other AppleTalk routes throughout the internet. The **`route show`** command is used to display the AppleTalk route table. For each route, the table lists:

- Destination network address.
- Network address of the next hop (if the route is to another router).
- Segment number associated with the next hop.
- Cost (number of hops, or intermediate routers).
- State (good, suspect, or bad).

Periodically, each AppleTalk router (including other PowerHubs serving as AppleTalk routers) broadcasts RTMP packets through each of its segments configured for AppleTalk to the other AppleTalk routers and nodes adjacent to it. As a result, each router in an AppleTalk network always has a current list of routes to the other networks. The syntax for this command is:

`route|rt [show] [-c|-r] [-t] [`*`<seglist>`*`] [`*`<net>`*`]`

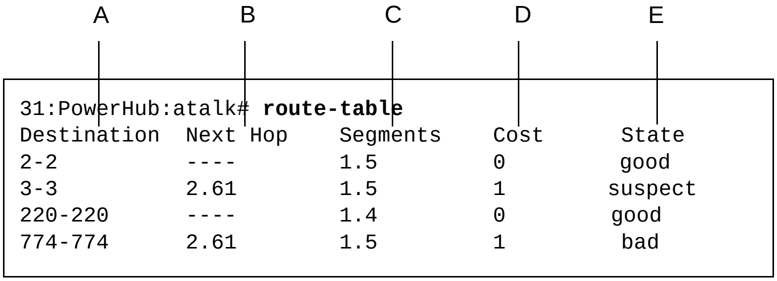
-c|-r Restricts the display to only directly connected entries (**-c**) or RTMP entries (**-r**).

-t Displays the total number of entries in the route table.

<seglist> Specifies the segments to display route information.

<net> Specifies the AppleTalk net address to display route information.

Here is an example of the use of the **`route show`** command.



```
31:PowerHub:atalk# route-table
Destination  Next Hop    Segments    Cost      State
2-2          ----        1.5         0          good
3-3          2.61        1.5         1        suspect
220-220      ----        1.4         0         good
774-774      2.61        1.5         1          bad
```

In this example, the routes for four destinations are shown:

A Lists the network address range for each route in the routing table.

B The network address of the router at the next hop. When a destination is local to the router, the next hop field contains dashes (----).

C Indicates the segment number through which the route can be reached.

D Indicates how many hops (routers) a packet must pass through to reach the destination.

E Lists the state of the route.

A route can have one of three states: good, suspect, or bad. Approximately every 10 seconds, the PowerHub broadcasts an RTMP packet to each adjacent PowerHub to inform of active (good) routes. When an RTMP packet is not received, within 20 seconds, it changes the status for the routes from good to suspect.

After a route becomes suspect, the PowerHub waits an additional 20 seconds to receive the status packet. If the packet is received within 20 seconds, the status is changed from suspect to good. If the packet is not received, the status changes from suspect to bad. When a route's status changes to bad, the PowerHub waits another 20 seconds for an RTMP packet. If the packet still is not received, the bad route is removed from the routing table.

Here is an example of the display produced if using the **-c** argument, which displays entries only for directly connected networks:

```
32:PowerHub:atalk# route show -c

Destination  Next Hop    Segments     Cost     State
2-2          --------     1.5          0       good
220-220      --------     1.4          0       good
```

Because the routes listed in this display are for directly connected destinations, no value appears under the Next Hop column for either route.

Here is an example of the display produced using the **-c** argument and specifying a specific segment:

```
33:PowerHub:atalk# route-table -c 1.4

Destination   Next Hop     Segments    Cost    State
220-220       ----          1.4         0      good
```

The argument used to produce this display restricts the information to only those routes that are directly connected and are attached to segment number 1.4.

NOTE

If the route table is blank, AppleTalk routing might not be enabled. Use the **config show** command to verify that routing is enabled.

19.7 Using the Route Cache

The AppleTalk route cache shows, for each segment, the most recently used destination networks. At any time an at-a-glance picture of AppleTalk routing activity in your network can be displayed by displaying the AppleTalk route cache.

19.7.1 Displaying the Route Cache

The **cache show** command is used to display the AppleTalk route cache. The syntax for this command is:

cache [show] [*<seglist>*]

<seglist> Specifies the segmentsto display information in the route cache. If no segment is specified, information for all segments is shown.

Here is an example of the display produced by this command:

```
33:PowerHub:atalk# cache show
Port 1.1:  empty
Port 1.2:  111.22,111.56
Port 1.3:  2.22
Port 1.4:  2.22
Port 1.5:  empty
Port 1.6:  empty
```

NOTE

The contents of the route cache can change quite rapidly. As a result, successive **cache show** commands can give different results.

19.7.2 Flushing the Route Cache

The **cache clear** command removes all entries for all segments from the route cache. After the cache is flushed, new entries are again added, using the cache's "most-recently-used" algorithm. Thus, the **cache clear** command can be used to ensure that all entries displayed by a subsequent **cache show** command are fresh.

19.8 Displaying NBP Information

The PowerHub uses Name Binding Protocol (NBP) to associate names with AppleTalk network numbers, node addresses, socket numbers, and other services. With NBP, a meaningful name can be bound to any service in an AppleTalk internet. For example, to bind the name “`Printer1`” to a socket number to which a printer is attached NBP could be used.

NOTE

The NBP table maintained by the PowerHub lists only the objects registered with the PowerHub.

For each service registered with the PowerHub , the NBP table lists:

- Object name.
- Object type.
- Zone in which the object resides.

To display the NBP table, use the **name show** command. Here is an example of the information displayed by this command:

```
34:PowerHub:atalk# name show
Object Name           ObjectType     Zone
PORT_220.150          Router         Macintosh
POWERHUB              Router         FORE Systems
```

A network administrator used AppleTalk NBP to name the two objects (services) “PORT_220.150” and “POWERHUB.” Both objects are registered to this PowerHub as type “Router.” They belong to different zones, “Macintosh” and “FORE Systems,” respectively.

19.9 Displaying Statistics

During operation of AppleTalk networks, the PowerHub collects statistics for AARP , Datagram Delivery Protocol (DDP), and AppleTalk Echo Protocol (AEP) packets. The **`stats` show** command is used to display statistics for AppleTalk ARP, DDP, or AEP packets. The syntax for this command is:

```
stats arp|ddp|echo [-t]
```

arp|ddp|echo Specifies the type of AppleTalk protocol to display statistics.

-t Displays statistics collected since the most recent switch reset, rather than those collected since the most recent clear (using the **stats clear** command).

The types of statistics the PowerHub collects and displays depends upon the protocol type. Here is an example of information displayed for the AARP protocol:

```
35:PowerHub:atalk# stats show arp
ARP Statistics:

Requests received:            992
Replies received:             296
Invalid packets received:     0
Requests sent:                79
Replies sent:                 0
Add arp entry failed:         0
```

Here is an example of the information displayed for the DDP protocol:

```
36:PowerHub:atalk# stats show ddp
DDP Statistics

Out Requests:                 93734
Out Shorts:                   0
Out Longs:                    93734
In Receives:                  82180
Forward Requests:             63849
In Local Datagrams:           78658
No Proto Handler:             0
Out No Routes:                0
Too Short Errors:             0
Too Long Errors:              0
Broadcast Errors:             0
Short DDP Errors:             0
Hop Count Errors:             0
Checksum Errors:              0
Config Address Errors:        0
Local Range Conflicts         0
Config Zone Errors:           0
Memory Allocation Errors      0
```

Here is an example of the information displayed for the AEP (echo) protocol:

```
37:PowerHub:atalk# stats show echo
Echo requests received:       39596
Echo replies received:        0
Echo requests sent:          0
```

NOTE

If a table displayed by the **stats** command contains all zeroes for the statistics amounts, AppleTalk routing might not be enabled. Use the **config show** command to verify that routing is enabled.

19.10Clearing AppleTalk Statistics

To clear the statistics collected since the most recent clear, use the **stats clear** command:

```
stats clear arp|ddp|echo
```

arp|ddp|echo Specifies the type of AppleTalk protocol to clear statistics.

19.11Testing a Network Address

The **ping** command can be used to test the accessibility of and round-trip delay to any AppleTalk node. This command sends an AEP packet to the specified node. The AEP packet contains an instruction to the receiving device to forward the packet back to the sending PowerHub, thus verifying receipt of the packet. To send an AEP packet, use the following command:

```
ping  [-t <timeout>] [-size <pktsize>] <net>.<node>
```

[-t <timeout>] Optionally specifies the number of seconds the PowerHub waits to receive a reply packet from the specified node. The default is **15** seconds.

[-size <pktsize>] If the <*timeout*> argument is used, optionally specifies the size of the echo packet to send to the node. The packet size is measured in bytes. Specify a packet size of 64-586 bytes. The default is **64** bytes.

<net>.<node> Specifies the network node to which to send the test packet.

The following example shows the results of the **`ping`** command when an AEP packet is successfully received by the sending PowerHub:

```
39:PowerHub:atalk# ping 220.150
220.150 is alive
```

If the target node to which an AEP packet is sent is not found, or if the timeout expires before the return packet is received, an error message is displayed.

In such a case, check the route table for the network on which the specified target node resides. If the network is listed in the table, check the configuration for the target node to ensure it has learned the current network and zone-related information. If the route table and target node are okay, check the physical connections between the PowerHub and the target node.

CHAPTER 20 Configuring IPX Routing

This chapter describes the commands in the ipx subsystem and tells how to use the commands to configure and manage the PowerHub as an IPX router. The commands in this subsystem to perform the following tasks:

- Allocate memory for IPX routing
- Show the switch's IPX configuration
- Add, show, and delete IPX interfaces
- Enable IPX routing
- Show, add, and delete IPX routes
- Show or clear the IPX route cache
- Configure IPX RIP
- Add, show, and delete IPX servers
- Configure IPX helper addresses
- Show and clear IPX statistics
- Customize the IPX routing behavior

20.1 Accessing the IPX Subsystem

To access the ipx subsystem, enter the following command from the runtime command prompt:

```
ipx
```

20.2 Allocating Memory for IPX Routing

Before the ipx subsystem can be used, memory must be allocated. Regardless of how much main memory the PowerHub contains, memory must be specifically allocated for use by the ipx subsystem.

NOTE

FORE Systems recommends that memory for the IPX subsystem be allocated immediately after booting the PowerHub to ensure that the memory requested is available. For more information, see the *PowerHub Hardware Reference Manual.*

To allocate memory for the `ipx` subsystem, issue the following command:

```
getmem
```

20.3 Showing the IPX Configuration

The current IPX settings can be displayed by issuing the **`config show`** command. Here is an example of the information displayed by this command:

```
6:GE:ipx# config show
IPX Configuration:

IPX Router:                   Memory Available
IPX Forwarding:               enabled
IPX Type20 Packet Forwarding: enabled
IPX Helper Feature:           enabled
Large RIP and SAP Packets:    disabled
RIP broadcast timer interval: 60
SAP broadcast timer interval: 60
RIP aging timer interval:     180
SAP aging timer interval:     180
You can set any of the IP configuration items listed in this display.
```

IPX Router Indicates whether main memory has been allocated for the IPX subsystem.

IPX Forwarding Indicates whether IPX forwarding is enabled or disabled. The default setting is disabled.

IPX Type20 Packet Forwarding Indicates that the switch is configured to forward type-20 IPX packets. The default setting is enabled.

IPX Helper Feature Indicates the setting of the IPX helper feature. When enabled, this feature allows the switch to forward unknown IPX broadcast packets.

Large RIP and SAP Packets Indicates whether the switch is enabled to forward large (greater than 576 bytes) IPX RIP and SAP packets. The default setting is disabled.

RIP broadcast timer interval Indicates how often RIP broadcasts are sent. Default is 60 seconds.

SAP broadcast timer interval Indicates how often SAP broadcasts are sent. Default is 60 seconds.

RIP aging timer interval Indicates how many seconds a learned, unused IPX route can remain in the route table before it is removed by the aging mechanism. Default is 180 seconds. If a value other than the default is selected, the RIP aging timer interval is always three times the RIP packet aging interval.

SAP aging timer interval Indicates how many seconds a learned, unused IPX server can remain in the server table before it is removed by the aging mechanism. The default is 180 seconds. If a value other than the default is selected, the SAP aging timer intervals always three times the SAP packet aging interval.

Any of the IPX configuration items listed in this display can be configured. Sections in this chapter describe the commands used to set these items.

20.4 Adding and Deleting IPX Interfaces

The **interface add** command is used to assign an IPX interface (sometimes referred to as a network number) to one or more PowerHub segments. When adding an interface, an entry is made in the route table to show that the network is directly connected to the specified segment. (See Section 20.7.1.) The syntax for the **interface add** command is:

```
interface|it add <segmentlist> <network>
    [mtu <mtu>] [met[ric] <metric>]
     [encap enet|802.3|802.2|snap]
```

<segmentlist> Specifies the segment number(s) assigned to the IPX interface. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

NOTE If more than one segment number per interface is specified, an IPX interface for a VLAN is created. Refer to *Chapter 15, Configuring IP Routing* for information on configuring VLANs.

<network> Specifies an IPX network number. Specify a hexadecimal number in the range from **`1`** through **`ffffffe`**.

[mtu <mtu>] Specifies the maximum transmission unit (number of octets) for packets forwarded on this segment. Specify a number in the range from **`576`** through **`1500`**. The default is **`576`**.

[met[ric <metric>] Specifies an additional cost (extra hops) of using the interface. Specify a cost in the range 1–14. When the PowerHub reports this subnet using RIP, it adds the additional cost to the reported metric. The default metric is 0.

Here are some examples of the use of this command.

```
1:PowerHub:ipx# interface add 1.2 1001 encap enet
Port 1.2, Network 0x1001, MTU 576, Cost 0, Frame type Ethernet II
Added
2:PowerHub:ipx# it add 1.6 2002 encap enet metric 1
Port 1.6, Network 2002, MTU 576, Cost 1, Frame type Ethernet II
Added
```

The first command creates an IPX interface on segment 1. Because this interface is intended to be used as the primary route to the PowerHub from a router, no cost is specified.

The second command creates an IPX interface on segment 6. However, a cost has been added to this interface. RIP adds this cost to the route when it reports it to the other routers attached to segment 6.

Here is an example of the **`interface add`** command used to add an IPX network to more than one PowerHub segment. This command creates an IPX VLAN.

```
23:PowerHub:ipx# it add 1.1, 1.2 55ccdd55 576 802.2
Port 1.1, Network 55ccdd55, MTU 576, Frame type 802.2
Added
Port 1.2, Network 55ccdd55, MTU 576, Frame type 802.2
Added
```

20.4.1 Deleting IPX Interfaces

The **`interface delete`** command is used to delete an IPX interface. The syntax for this command is:

`interface|it del[ete]` ***`<segmentlist>`*****`|all`** ***`<network>`*****`|all`**

<segmentlist>|all Specifies the segment(s) to delete. If **`all`** is specified, the network number is removed from all the PowerHub segments.

<network>|all Specifies the IPX network to delete. If **`all`** is specified, all IPX networks are deleted from the specified segment(s).

20.5 Displaying IPX Interfaces

Network numbers assigned to segments can be viewed by using the **`interface show`** command. The syntax for this command is:

`interface|it [show]` ***`<segmentlist> <network>`***

<segmentlist> Specifies the segments to display IPX interface information. If a list or range of segments is specified, information is shown for only those segments that have IPX interfaces.

<network> Specifies the IPX network for which to display information.

The display includes the segment state—UP, if the segment is up, or DOWN, if the segment is disabled or if the automatic segment-state detection mechanism has determined the segment to be down. An example of the information displayed by this command.

```
25:PowerHub:ipx# interface show
 Port Network Address  MTU  Encapsulation  State  Cost
 ---- ---------------  ---  -------------  -----  ----
 1.1   00001001        576    enet          UP     0
 1.1   55ccdd55        576    802.2         UP     0
 1.2   55ccdd55        576    802.2         UP     0
 1.3   55ccdd55        576    802.2         UP     0
 1.6   00002002        576    enet          UP     1
```

20.6 Enabling IPX Routing

Enable IPX forwarding after defining the IPX interfaces (see Section 20.4). By enabling IPX forwarding, the IPX software can send and receive RIP and SAP updates, and respond to RIP and SAP requests from stations. Use the following command to enable IPX forwarding:

```
[ipx] enable|disable
```

enable|disable Specifies whether enabling or disabling IPX forwarding. The default state of forwarding is disabled.

20.6.1 Adding and Deleting IPX Routes

Use the **route add** command to assign the route to be used when forwarding to a particular network. The syntax for this command is:

```
route|rt add <network> <gw-net> <gw-addr> <seg> <hops> <ticks>
```

<network> Specifies the destination IPX network number, a 32-bit value expressed as up to eight hexadecimal digits. Specify a number in the range from **1** through **fffffffe**.

<gw-net> Specifies the network number of the gateway (IPX router) through which packets for the destination network are to be routed. This network number must be one of the network numbers that is already configured on the segment specified by the <*seg*> argument. Specify a number in the range from **1** through **fffffffe**.

<gw-addr> Specifies the IPX node number of the gateway (router) to which packets for the destination network should be forwarded. An IPX node number is actually a 48-bit MAC-layer address. Such an address is expressed in PowerHub commands as six hexadecimal bytes separated by hyphens.

The gateway should be a device connected to a network that is directly attached to the PowerHub segment specified in the <*seg*> argument.

<seg> Specifies the PowerHub segment on which a packet should be forwarded to reach the specified gateway and, eventually, the specified network.

<hops> Specifies the number of hops to the destination, that is, how many gateways a packet must go through to reach the specified network.

A hop-count of **1** corresponds to a direct connection. (Note, however, that you cannot add a route to a network that is directly attached.)

The maximum number of hops is **15**; a hop-count of **16** is synonymous with "infinity" and means that the specified network is unreachable.

<ticks> Specifies the typical delay expected for a packet to reach its destination, measured in 55-mS "ticks."

In Ethernet, FDDI, and other networks with bandwidths greater than 1 Mb/s, each network is assumed to create a delay of one tick. If a route includes only such networks, the number of ticks should be set equal to the number of network segments in the route, which is the number of hops plus 1. However, routing paths that include slow, wide-area links (ex: 56 Kb/s leased lines) should have a larger number of ticks to account for the slow links.

Ticks are represented in IPX by 16-bit integers, so the practical maximum number of ticks is far less than the number that can be entered here.A statically-entered IPX route is always marked as "UP" when it is added. The route is automatically marked as "DOWN" when the corresponding segment is disabled, either manually in the `bridge` subsystem or automatically by the automatic segment-state detection mechanism.

When routing a packet to a remote network, the IPX routing software selects the route with the lowest number of ticks, regardless of whether it is a static route or a dynamic route. When two or more routes to a remote network have an equal number of ticks, the router chooses the route with the smallest number of hops. An example of the **route add** command is shown below:

```
7:PowerHub:ipx#rt add 008ffff9 96aabb69 0-0-99-88-88-8 2.32 3
Route to 008ffff9 via 96aabb69: added.
```

The result of this command is that packets directed to network 008ffff9 are forwarded on segment 2.3 to a gateway with address 0-0-99-88-88-88, and can expect to require a total of 2 hops and 3 ticks to reach a station on the destination network.

20.6.2 Deleting IPX Routes

Static routes can be completely eliminated using the **`route del`** command. The syntax for this command is:

```
route|rt del[ete] <network> <gw-net> <gw-addr>
```

<network> Specifies the destination IPX network number, a 32-bit value expressed as up to eight hexadecimal digits.

<gw-net> Specifies the network number of the gateway (IPX router).

<gw-addr> Specifies the IPX node number of the gateway (router).

20.6.3 Displaying IPX Routes

The **`route show`** command is used to display the IPX route table. The syntax for this command is:

```
route|rt [show] [-c|-r|-t] <seglist> <network>
```

-c|-r|-t Restricts the display to one of the following:

-c Only directly connected entries

-r Only remotely attached entries

-t Displays the total count of UP and DOWN routes.

<seglist> Specifies the segment(s) to display route information.

<network> Specifies the IPX network to display route information.

Here is an example of the display produced by this command:

```
60:PowerHub:ipx# route show

        Destnet   Gway-net  Gway-nodeaddr     Hops  Ticks  State  Age Sgmts
        --------  --------  ----------------  ----  -----  -----  --- -----
        00001001  --------  ------------        1     2     UP    ---   1
        00002002  --------  ------------        1     2     UP    ---   6
        55ccdd55  --------  ------------        1     2     UP    ---   1
        55ccdd55  --------  ------------        1     2     UP    ---   2
        55ccdd55  --------  ------------        1     2     UP    ---   3
        008ffff9  96aabb69  00-00-99-88-88-88   2     3     UP    ---   8
        054ffff9  f4f4f4f4  00-00-99-22-22-22   2     3     UP    ---   4
        064ffff9  f4f4f4f4  00-00-99-22-22-22   2     4     UP    ---   4
        011ffff9  96aabb69  00-00-99-11-11-11   2     3     UP    ---   3
        165ffff9  00fabcab  00-00-99-44-44-44   2     3     UP    ---   9

Total no. of routes = 10 (10 UP, 0 DOWN)
```

This command displays the following information about IPX routes:

Destnet IPX network number of the destination network.

Gway-net If the destination is not directly attached, this field contains the IPX network number of the gateway (IPX router) through which packets for the destination are to be routed.

Gway-nodeaddr If the destination is not directly attached, this field contains the node address of the IPX gateway (router) through which packets for the destination are to be routed.

Hops The number of gateways, including the PowerHub, that a packet must go through to reach the destination. If a network is directly attached, the hop-count is 1.

Ticks The number of 55-mS ticks that can be expected for a packet to reach its destination. If all of the network segments along the route have a bandwidth of 1 Mb/s or more, the number of ticks generally equals the number of hops plus 1. Otherwise, it is larger to account for the slower segments.

State The state of the route; UP or DOWN. When a segment goes down, its state is updated in the interface table. All routes that use this segment are marked DOWN in the route table, and all servers that are not accessible except through this segment are marked as DOWN in the server table.

When the segment comes back up, its state is again updated in the interface table. All routes that use this segment are marked as UP in the route table, and servers that are now accessible through this segment are marked as UP in the server table.

Age For dynamic routes, the number of seconds that have elapsed since this routing information was received. The Age field displays "---" for direct/static routes. For RIP entries, the Age field displays how long it has been since a routing update for the route has been received.

Ports Lists the segments on which packets for this destination should be forwarded.

The software does not contain a command to directly take a static route DOWN. To take DOWN a static route, use the **`route delete`** command to remove the route.

20.7 Displaying and Clearing the IPX Route Cache

The IPX route cache shows, for each segment, the most recently used destination networks. At any time, an at-a-glance picture of IPX routing activity in your network can be displayed by displaying the IPX route cache.

20.7.1 Displaying the Route Cache

The **`cache show`** command is used to display the IPX route cache. The syntax for this command is:

```
cache [show] <seglist>
```

<seglist> Specifies the segments to display information in the route cache. If a segment is not specified, information for all segments is shown.

Here is an example of the output produced by this command. The cache displayed in this example is for a PowerHub containing 14 segments.

```
66:PowerHub:ipx# cache show
IPX router cache:
Segment 1.1: empty
Segment 1.2: empty
Segment 1.3: 011ffff9, 96aabb69
Segment 1.4: f4f4f4f4, 054ffff9, 064ffff9
Segment 1.5: empty
Segment 1.6: empty
Segment 2.1: empty
Segment 2.2: 00000022
Segment 2.3: 00fabcab, 165ffff9
Segment 2.4: empty
Segment 2.5: empty
Segment 2.6: empty
Segment 3.1: empty
Segment 3.2: empty
```

NOTE

The contents of the route cache can change quite rapidly. As a result, successive **cache show** commands can give different results.

20.7.2 Clearing the Route Cache

The **cache clear** command removes all entries from all segments in the route cache. After the cache is flushed, new entries are again added, using the cache’s “most-recently-used” algorithm. Thus, the **cache clear** command can be used to ensure that all entries displayed by a subsequent **cache show** command are fresh.

In the following example, the route cache is flushed once and then quickly displayed two times.

```
67:PowerHub:ipx# cache clear
IPX router cache flushed
68:PowerHub:ipx# cache show
IPX router cache:
Segment 1.1: empty
Segment 1.2: empty
Segment 1.3: 96aabb69
Segment 1.4: f4f4f4f9
Segment 1.5: empty
Segment 1.6: empty
Segment 2.1: empty
Segment 2.2: empty
Segment 2.3: empty
Segment 2.4: empty
Segment 2.5: empty
Segment 2.6: empty
Segment 3.1: 00001fd1
Segment 3.2: empty
69:PowerHub:ipx# cache show
IPX router cache:
Segment 1.1: empty
Segment 1.2: empty
Segment 1.3: 96aabb69, 011ffff
Segment 1.4: f4f4f4f4, 054ffff
Segment 1.5: empty
Segment 1.6: empty
Segment 2.1: empty
Segment 2.2: 00000022
Segment 2.3: 00fabcab
Segment 2.4: empty
Segment 2.5: empty
Segment 2.6: empty
Segment 3.1: 00001fd1
Segment 3.2: empty
```

20.8 Configuring IPX RIP and SAP Parameters

Earlier sections in this chapter describe how to add static entries to the IPX RIP and SAP tables maintained by the PowerHub. However, the software contains additional RIP and SAP options that can be configured:

- Whether updates are generated on a per-segment basis or a per-VLAN basis.
- Whether large (greater than 576 bytes) IPX RIP and SAP packets are generated.
- Talk and listen (send and receive) settings for each interface or segment.

20.8.1 Setting the Control Type

The RIP and SAP control type can be set to change the RIP and SAP update mechanism. Using the **`set ripsap-ctrl`** command, the PowerHub can be configured to generate and send a copy of each RIP and SAP packet on a per-VLAN basis instead of on a per-segment basis.

If the IPX configuration does not contain IPX VLANs, PowerHub performance can be the same whether configured to generate updates on a per-segment basis or a per-VLAN basis. In this case, it is recommended that the configuration be left in its default state: generate updates on a per-segment basis.

However, if the configuration does include IPX VLANs, performance can be enhanced by configuring the software to use the per-VLAN method for generating the RIP and SAP updates. When the control type is changed to **`vlan`**, the software spends less time generating RIP and SAP updates, because it generates only a single update for each network, even if the network spans multiple segments. To change the RIP and SAP update method, issue the following command:

```
set ripsap-ctrl|rsct [normal|n vlan|v]
```

normal|n Specifies that RIP and SAP updates are generated on a per-segment basis. This is the default.

vlan|v Specifies that RIP and SAP updates are generated on a per-VLAN basis.

If no parameter is used with this command, the current control type is displayed.

NOTE This command affects only IPX RIP and SAP updates. It has no affect on IP RIP updates.

20.8.1.1 Displaying the RIP and SAP Control Type

To display the RIP / SAP control type, issue the following command:

```
ripsap-ctrl|rsct [show]
```

Here are the results produced by this command:

```
399:PowerHub:ipx# ripsap-ctrl show
ripsap-ctrl-type:      normal
```

20.8.1.2 Adjusting the Interval and Aging Timers

The RIP and SAP timers can be adjusted. The PowerHub IPX implementation generates and transmits RIP and SAP updates at regular intervals. RIP updates contain information about the IPX routes known to the PowerHub. SAP updates contain information about the UPX servers known to the PowerHub.

The default interval for RIP and SAP updates is 60 seconds. Every 60 seconds, IPX RIP and SAP updates are generated and transmited. Depending on whether RIP and SAP updates are configured to use the per-segment method or the network method, updates are generated for each segment or for each network.

Aging is a mechanism that periodically clears learned entries from the RIP and SAP tables. At a specified interval (the aging interval) the PowerHub determines which of the learned entries in the table have not been recently used. For proper RIP and SAP reporting, the aging interval must be at least three times the duration of the broadcast interval. If an entry is not used during the specified interval, it is discarded. A separate broadcast interval and aging timer are maintained for IPX RIP and for IPX SAP. To set interval and aging timers for RIP, issue the following command:

```
timers set <transmit-intvl> [<rip-age>]
```

<transmit-intvl> Sets the RIP broadcast interval. Specify a value from **60** to **600** seconds. The default is **60** seconds.

<rip-age> Sets the RIP age timer. If specified, the RIP age timer value must be at least three times the value of the RIP broadcast interval. Specify a value between **180** and **1800** seconds. If unspecified, this argument defaults to three times the value of the RIP broadcast interval.

Here is an example of this command:

```
400:PowerHub:ipx/rip# timers 100 300
```

To set interval and aging timers for SAP, issue the following command:

```
timers set <transmit-interval-time> [<aging-time>]
```

<transmit-interval-time> Sets the SAP broadcast interval. Specify a value from **60** to **600** seconds. The default is **60** seconds.

<aging-time> Sets the SAP age timer. If specified, the SAP age timer value must be at least three times the value of the SAP broadcast interval. Specify a value between **180** and **1800** seconds. IF unspecified, this argument defaults to three times the value of the SAP broadcast interval.

Here is an example of this command:

```
400:PowerHub:ipx/sap# timers 100 300
```

20.8.2 Setting Talk and Listen for RIP and SAP

The following sections describe the commands available in the ipx/rip subsystem for setting and disabling talk and listen parameters for RIP and SAP.

20.8.2.1 Setting RIP Parameters

To enable IPX RIP sending (**talk**) or receiving (**listen**), use the **talk penable** and **listen penable** commands in the `ipx/rip` subsystem. The syntax for these commands is:

```
talk|ta penable <seglist>|all
talk|ta nenable <network>
listen|li penable <seglist>|all
listen|li nenable <network>
```

<seglist>|all Specifies the segments to be enable IPX RIP sending or receiving. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. If **all** is specified, IPX RIP is enabled for all segments.

<network> Specifies the following:

talk|ta Enables the sending of RIP update packets to the specified network.

listen|li Enables the learning of routes from RIP packets received from the specified network.

20.8.2.2 Disabling RIP Parameters

To disable IPX RIP **talk** or **listen**, use the **talk pdisable** and **listen pdisable** commands in the `ipx/rip` subsystem. Here is the syntax for these commands:

```
talk|ta pdisable <seglist>|all
talk|ta ndisable <network>
listen|li pdisable <seglist>|all
listen|li ndisable <network>
```

20.8.2.3 Setting SAP Parameters

To enable IPX SAP sending (**talk**) or receiving (**listen**), use the **talk penable** and **listen penable** commands in the `ipx/sap` subsystem. Here is the syntax for these commands:

```
talk|ta penable <seglist>|all
talk|ta nenable <network>
listen|li penable <seglist>|all
listen|li nenable <network>
```

<seglist>|all Specifies the segments for which IPX SAP sending or receiving is set. Specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. If **all** is specified, IPX SAP is enabled for all segments.

<network> Specifies the following:

talk|ta Enables the sending of SAP update packets to the specified network.

listen|li Enables the learning of routes from SAP packets received from the specified network.

20.8.2.4 Disabling SAP Parameters

To disable IPX SAP **talk** or **listen**, use the **talk pdisable** and **listen pdisable** commands in the `ipx/sap` subsystem. Here is the syntax for these commands:

```
talk|ta pdisable <seglist>|all
talk|ta ndisable <network>
listen|li pdisable <seglist>|all
listen|li ndisable <network>
```

20.8.3 Displaying the Configuration

To display the talk and listen (send and receive) settings for RIP and SAP updates, use the **config show** command in both the `ipx/rip` and `ipx/sap` subsystems within the `ipx` subsystem. Here is the syntax for this command.

config show [*<seglist>*] [*<network>*]

<seglist> Specifies the segments to display IPX RIP and IPX SAP configurations. If no segment is specified, all RIP and SAP control table entries are displayed.

<network> Network address of the network to display RIP and SAP control table entries. If no network is specified, all RIP and SAP control table entries are displayed.

Here is an example of the display produced by this command if **normal** was selected in the **set ripsap-ctrl** command.

```
91:PowerHub:ipx/rip# config show

Segment       Talk      Listen
-------       ----      ------
 1.1             yes       yes
 1.2             yes       yes
 1.3             yes       yes
 1.4             yes       yes
 1.5             yes       yes
 1.6             yes       yes
```

20.8.4 Setting the Parameters

The IPX software advertises and receives IPX routing information using the IPX RIP . IPX server information is advertised and received using the SAP.

NOTE

The RIP protocol used by IPX is different from RIP used in IP.

The commands for displaying the talk and listen (send and receive) settings for IPX RIP and SAP differ depending upon the update method used by the software:

- If the update method is per-segment, use the **`penable`** command in both the `ipx/rip` and `ipx/sap` subsystems within the `ipx` subsystem.
- If the update method is per-VLAN, use the **`nenable`** command in both the `ipx/rip` and `ipx/sap` subsystems within the `ipx` subsystem.

20.8.5 Equal RIP Route

To enable or disable accepting the first equal RIP route to the network, issue the following command:

```
one-rip-entry|onere enable|disable
```

20.9 Using the Server Table

Information about NetWare file servers and other NetWare services are stored in a data structure called the server table. The IPX routing software maintains a server table containing information that it uses when advertising services and responding to server information requests using SAP. The table contains two types of servers:

Dynamic servers Learned by the switch through the SAP. IPX file servers, print servers, and other service providers advertise their existence using SAP. This information is learned by all IPX routers in the network. When an IPX station requires a service, it uses SAP to request server information from the nearest router.

Static servers Configured by a system administrator, using the **`server add`** command. The IPX routing software always has SAP enabled, and services are always being discovered and advertised dynamically. Although the information learned through SAP is usually sufficient for good network behavior, there might be occasions in which you would like to make permanent entries in the server table. For example, permanent entries can be made in the server table to ensure quick availability of service information after a network outage. Static service assignments can be used for this purpose.

NOTE

Before adding a server to the IPX server table, a route (to the IPX route table) must be added to the server's net.

When responding to IPX stations' requests for the information on the "nearest" server of a given type, the PowerHub selects the server with the best route as determined from the route table, regardless of whether the server is static (added to the server table permanently by the **server add** command) or dynamic (learned through SAP). If there are equally good routes to two or more servers, the software chooses the server with the least number of hops in the server table.

20.9.1 Displaying the Server Table

To display known IPX servers, issue the following command:

```
server [show] [-f|-a|-t]
<seglist> <network> <name> <type>
```

[-f|-a|-t] Specifies the type of entries to display:

- -f Displays the entire server name, up to 48 characters. Otherwise, a maximum of 24 characters is displayed to keep the display within an 80-character line.
- -t Displays only the total count of UP and DOWN server entries.
- -a Displays the network number and MAC address of the next-hop gateway.

<seglist> Specifies the segment(s) to display route information.

<network> Specifies the IPX network number of the server.

<name> If a server name is specified, only information that applies to the specified server is displayed.

<type> Specifies the type of service, either a mnemonic or a 16-bit number in the range **0** through **fffe**, expressed as up to four hexadecimal digits:

Table 20.1 - Service Types

Mnemonics	Server-type(hex)
PRINT-QUEUE	0003
FILE-SERVER	0004
JOB-SERVER	0005
PRINT-SERVR	0007
ARCHIVE-SVR	0009
REM-BRIDGE	0024
ADVRT-PRINT	0047

Here is an example of the output produced by this command.

```
73:PowerHub:ipx# server show
SrvrTyp  SrvrNet  SrvrNode          Sock  Hop  State Sgmt  Age  Srvr-name
        ------- -------- -----------------  ----  ---  ----- ----  ---  ---------
        00ff    f4f4f4f4 00-00-99-66-66-66  f4f4  2     UP    4         crp-srvr
        00fe    f4f4f4f4 00-00-99-66-66-66  f4f4  2     UP    4         big-boss
        0123    f4f4f4f4 00-00-99-66-66-66  f4f4  2     UP    4         cat-mkr
Total no. of servers = 3 (3 UP, 0 DN)
```

This command displays the following information from the server table:

Server-type Specifies the type of service, either a mnemonic or a 16-bit number in the range **`0`** through **`fffe`**, expressed as up to four hexadecimal digits.

Srvr-net The IPX network number of the server.

Server-node The IPX node number of the server.

Sock The IPX socket number on which the server accepts requests for service.

Hop The number of gateways, including the PowerHub, that a packet must go through to reach the server. If the server is on a directly-attached network, the hop-count is 1.

State This is the state of the server; possible states are “UP” and “DOWN.”

Segment The segment on which the entry was learned.

Age For dynamic servers, the number of seconds that have elapsed since this information was received.

Server-name The name of the server, up to 48 ASCII characters.

20.9.2 Adding a Static Server

To add a server to the server table, use the **`server add`** command. Here is the syntax for this command.

```
server add <s-type> <s-net> <s-addr> <s-sock> <s-hops> <s-name>
```

<s-type> Specifies the type of service, either a mnemonic or a 16-bit number in the range **`0`** through **`fffe`**, expressed as up to four hexadecimal digits (see Table 20.1)

<s-net> Specifies the IPX network on which the server resides, a 32-bit number expressed as up to eight hexadecimal digits.

Note that the PowerHub does not accept the **`server add`** command if there is no known route to the server's network at the time the command is given. Specify a number in the range from **`1`** through **`ffffffe`**.

<s-addr> Specifies the IPX node number of the server. This is a 48-bit MAC-layer address, expressed as six hexadecimal bytes separated by hyphens.

<s-sock> Specifies the IPX socket number on which the specified server accepts requests for service.

<s-hops> Specifies the number of hops to the specified server, that is, how many gateways a packet must go through to reach it. The maximum number of hops is **`15`**; a hop-count of **`16`** is synonymous with "infinity" and means that the specified server is unreachable.

<s-name> Specifies the name of the server, up to 48 ASCII characters. Server names are case sensitive.

Here is an example of the **`server add`** command:

```
3:PowerHub:ipx# server add 4 fabcab 0-0-88-88-88-88 1010 2 phsrvr
Server phsrvr of type 0004 on net 00fabcab: added.
```

20.9.3 Deleting a Static Server

A static server assignment can be deleted by using the **server delete** command. Here is the syntax for this command.

server del[ete] ***<s-type>*** **n[ame]** ***<s-name>***

<s-type> Specifies the type of service, either a mnemonic or a 16-bit number in the range **0** through **fffe**, expressed as up to four hexadecimal digits (see Table 20.1)

<s-name> Specifies the name of the server, up to 48 ASCII characters. Server names are case sensitive.

Here is an example of the use of this command.

```
72:PowerHub:ipx# server del 4 eng-server
Server eng-server of type 0004: deleted from table.
```

20.10Using IPX Helper

This section describes how to use the IPX Helper feature. IPX Helper lets the PowerHub forward unknown IPX broadcast packets, which normally would be dropped, onto specified networks. This feature forwards the unknown IPX broadcast packets without using the IPX SAP protocol. When an IPX helper address is assigned to a segment, and an unknown IPX broadcast packet with the specified destination socket number is received on that segment:

- The IPX broadcast packet destination network number and destination node address are replaced with the number and address specified in the **helper add** command.
- The IPX broadcast packet then is forwarded onto all other segments.

To use IPX Helper, you first must enable it by issuing the following command:

enable|enl disable|dis helper

20.10.1 Adding an IPX Helper Address

The **helper add** command is used to add an IPX helper address to a segment. Here is the syntax for this command:

```
helper add <seglist> <network> <node address> <socket>
```

<seglist> Specifies a segment, a comma-separated list of segments, or a hyphen-separated range of segments.

<network> Specifies a network number or the value **ffffffff** to specify all net broadcast.

<node address> Specifies the unicast address or the broadcast address **ff-ff-ff-ff-ff**.

<socket> Specifies a socket number in hexadecimal notation. To specify any socket number, enter the value **ffff**.

Here is an example of how to add an IPX Helper address. In this example, a broadcast address is defined.

```
95:PowerHub:ipx# helper add aabbccdd ff-ff-ff-ff-ff-ff ffff 1
```

20.10.2 Displaying an IPX Helper

The **helper show** command is used to display IPX helper addresses assigned for all segments. Here is an example of the information displayed by this command.

```
220:PowerHub:ipx# helper show

SEGMENT   NETWORK    NODE ADDRESS         SOCKET NUMBER
-------   -------    ------------         -------------
   1      aabbccdd   ff-ff-ff-ff-ff-ff    ffff
```

20.10.3 Deleting an IPX Helper Address

The **helper delete** command is used to delete an IPX helper address assigned to a segment. The syntax for this command is:

```
helper delete <seglist>
```

<seglist> Specifies a segment, a comma-separated list of segments, or a hyphen-separated range of segments.

20.11Showing and Clearing Statistics

The **stats** command is used to display IPX or type-20 packet statistics. Here is the syntax for this command:

stats [show] [-t]

-t Optionally displays statistics collected since the most recent switch reset, rather than those collected since the most recent clear.

Here is an example of the output produced by the **stats** command.

```
80:PowerHub:ipx# stats
IPX statistics: count since last stats clear
Datagrams received:                 2302091
Header errors received:             0
Address errors received:            0
Datagrams forwarded:                2302091
Unknown Broadcast packets forwarded: 0
Unknown protocols received:         0
Incoming datagrams discarded:       0
Datagrams delivered to higher layer: 2258
Datagrams sent:                     6658
```

Here is an example of the output produced by the **stats type20** command.

```
81:PowerHub:ipx# t20stats
Type-20 statistics: count since last stats clear
Packets   received:                 0
Packets   forwarded:                0
Packets   discarded:                0
Packets   in error:                 0
```

Here is an example of the use of the **-t** argument with the **stats** command. In this example, IPX statistics collected since the last switch reset are displayed.

```
83:PowerHub:ipx# stats -t
IPX statistics: Total count since last system reset
Datagrams received:                 2305309
Header errors received:             0
Address errors received:            0
Datagrams forwarded:                2305309
Unknown Broadcast packets forwarded: 0
Unknown protocols received:         0
Incoming datagrams discarded:       0
Datagrams delivered to higher layer: 2261
Datagrams sent:                     6664
```

To clear statistics, use the **stats clear** command.

20.12Customizing the IPX Configuration

To enable or disable the forwarding of type-20 packets for the entire switch, issue the following command:

```
enable|disable type20-forwarding|t20fw
```

20.12.1 Type-20 Forwarding for Segments

The **`type20-port-forwarding`** command is used to show whether type-20 packet forwarding is enabled or disabled on specific segments. The syntax for this command is:

```
penable|pdisable type20-port-forwarding|tpfw <seglist>
```

penable|pdisable Specifies whether type-20 packet forwarding is to be enabled or disabled. The default **`type20-port-forwarding`** is enabled.

<seglist> Specifies a segment, a comma-separated list of segments, or a hyphen-separated range of segments for which to enable or disable type-20 packet forwarding.

20.12.2 Enabling Large Packets

In software version 3.0, IPX RIP and IPX SAP packets larger than 576 bytes (the default minimum) can be generated. To change the default, use the **`enable large-rip-sap-pkt`** command to enable the software to generate large RIP and SAP packets. The syntax for this command is:

```
enable|disable large-rip-sap-pkt|lpkt
```

enable|disable Specifies that the PowerHub generate or not generate IPX RIP or IPX SAP packets larger than 576 bytes. The default for **`large-rip-sap-pkt`** is disabled.

NOTE The MTU setting for the IPX interfaces defined on the switch needs to be more than 576 bytes to generate larger RIP and SAP packets.

CHAPTER 21 Configuring IPX Translation Bridging

IPX translation bridging allows one or more IPX networks that span across FDDI and Ethernet segments to be configured using different packet encapsulations. Without altering the configurations of individual devices, IPX translation bridging enables Ethernet and FDDI devices with different encapsulation types to communicate with each other. This feature is especially useful if the IPX network consists largely of Ethernet devices using 802.3 encapsulation, the default encapsulation type in Novell IPX software versions 2.2 through 3.11.[1] However, if the network name is not in the IBT table, IPX translation bridging does not occur, and normal bridging does. This section describes how to perform the following tasks:

- Show the switch's IPX translation-bridging configuration
- Add, show, and delete IPX translation-bridging interfaces

NOTE

IPX translation bridging is independent of IPX routing—they are mutually exclusive. It is recommended that both IPX translation bridging and IPX routing not be enabled. However, if both IPX translation bridging and IPX routing are enabled, IPX routing takes precedence over IPX translation bridging.

21.1 Encapsulation Types

When IPX translation bridging is used, the Ethernet and FDDI encapsulation types to be used on each IPX network are specified. For each IPX network number, both the Ethernet and FDDI encapsulations to be used on that network can be specified. Table 21.1 lists the combinations of encapsulation types you can specify.

[1] If the FDDI device does not support 802.3 and it cannot bridge between Ethernet devices and the FDDI device using standard IPX bridging. Use IPX Translation bridging in this case.

Table 21.1 - Encapsulation Types

	ENET	802.2	802.3	SNAP
FDDI	4	4	4	4
Ethernet	4	4	4	4

NOTE

FDDI "raw" encapsulation is 802.3-like and is listed as "802.3" in table and command descriptions. However, this encapsulation is not identical to the 802.3 format on Ethernet since it does not include an explicit length field.

21.2 Configuration Requirements

Although IPX translation bridging is simple to configure, the following conditions must be met:

- The servers attached to the segments in an IPX translation bridging network must be configured to have the same network number as the "IPX translation-bridging" network number configured on the PowerHub. If a server's network number cannot be changed to correspond to the IPX translation-bridging network defined on the switch, change the network number to match the server.
- Servers and clients must be configured to have the same encapsulation type as the type specified for the appropriate medium in the IPX translation-bridging network. For example, a client attached to an Ethernet segment must be configured to use the same Ethernet encapsulation type as the one defined for the corresponding IPX translation-bridging network. However, if encapsulation types on the server or client cannot be changed, the encapsulation types of the client or server can be configured on the switch.

21.2.1 Enabling IPX Translation Bridging

Before the IPX translation-bridging feature can be used, IPX translation bridging must be enabled. To enable IPX translation bridging, issue the following command:

```
ipx-br-translation|ibt enable|disable
```

enable|disable Specifies whether IPX translation bridging is to be enabled or disabled.

Here is an example of the use of this command:

```
1:PowerHub:bridge# ipx-br-translation enable
IPX translation bridging is now enabled
```

21.2.2 Adding IPX Translation-Bridging Interfaces

To create an IPX translation-bridging network, use the following command:

```
ipx-br-translation|ibt add <network> <ether-encap> <fddi-encap>
```

<network> Specifies the IPX network number to apply the encapsulation settings.

<ether-encap> Specifies the encapsulation type to be used for Ethernet packets. Packets bridged from FDDI to this network number are converted to this encapsulation. Specify one of the following:

enet	Ethernet Type II encapsulation.
802.3	Raw 802.3 encapsulation.
802.2	802.3 with an LLC header.
snap	802.3 with LLC and SNAP headers.

NOTE The default Ethernet encapsulation type used in Novell IPX versions 2.2 through 3.11 is 802.3. The default for versions 3.12 through 4.x is 802.2.

<fddi-encap> Specifies the encapsulation type to be used for packets translated to FDDI. Specify one of the following:

802.3	Raw 802.3 encapsulation.
802.2	802.3 with an LLC header.
snap	802.3 with LLC and SNAP headers.

NOTE

The default FDDI encapsulation type used in Novell IPX versions 2.2 through 3.11 is 802.3, the same type used for Ethernet devices. Similarly, the default FDDI encapsulation type used in versions 3.12 through 4.X is 802.2.

Here are some examples of how to use this command. In these examples, definitions are created for IPX translation-bridging networks 100, 200, and 300:

```
2:PowerHub:bridge# ipx-br-translation add 100 802.2 snap
IPX network 100 added to the translation table
3:PowerHub:bridge#  ipx-br-translation add 200 802.2 802.2
IPX network 200 added to the translation table
4:PowerHub:bridge#  ipx-br-translation add 300 802.3 snap
IPX network 300 added to the translation table
```

21.2.3 Displaying IPX Translation-Bridging Interfaces

Definitions for the IPX translation-bridging networks defined on the PowerHub can be displayed at any time. To display the definitions, use the following command:

ipx-br-translation|ibt show [*<network>*]|[-t]

<network> Specifies an IPX translation-bridging network number.

-t Displays only the total number of entries in the IPX translation-bridge table.

Here are some examples of displays produced by this command. In the first example, no specific network number is given, so all individual entries are displayed, as well as the total number of entries. In the second example (prompt 7), the **-t** argument is used to display the total number of IPX translation-bridging entries.

```
6:PowerHub:bridge# ipx-br-translation show
IPX Translation Bridging: Enabled
IPX Network        Ethernet Encap        FDDI Encap
-----------        --------------        ----------
        100        802.2                 802.2/SNAP
        200        802.2                 802.2
        300        Ethernet II           802.2/SNAP

Total entries:  3
7:PowerHub:bridge# ipx-br-translation show -t
IPX Translation Bridging: Enabled
IPX Network        Ethernet Encap        FDDI Encap
-----------        --------------        ----------
Total entries:  3
```

21.2.4 Deleting IPX Translation-Bridging Interfaces

To delete the encapsulation settings assigned to a network number, use the following command:

ipx-br-translation|ibt del *<network>*|all

<network>|all Specifies an IPX translation-bridging network number. If **all** is specified, all IPX translation-bridging networks are deleted.

Here is an example of the use of this command:

```
8:PowerHub:bridge# ipx-br-translation del all
All IPX networks deleted from the IPX translation table
```

CHAPTER 22 Configuring DECnet Routing

The PowerHub contains a complete set of DECnet Phase IV routing software for use in DECnet networks. The routing engine works side-by-side with the Ethernet bridging software. With appropriate configuration, the PowerHub can be set up to perform DECnet routing on any segments.

This chapter assumes a familiarity with the basic requirements of DECnet networks and the DECnet protocol. For further information on this subject, refer to a DECnet guide, such as the DECnet Phase IV General Description, Order No. AA-N149A-TC, (Digital Equipment Corporation, 1982).

This chapter describes the commands and facilities of the DECnet subsystem. To set up the PowerHub for DECnet routing, the following steps must be performed:

1. Allocate memory for DECnet routing.
2. Assign the DECnet node ID using the **set node-id** command.
3. If the PowerHub is to be a Level-2 router, select it with the **set node-type** command.
4. Turn on DECnet routing with the **enable dec** command.
5. Enable DECnet routing on the desired segments with the **penable dec** command.

A large number of nodes may necessitate increasing the maximum limits for these parameters with the **set max-area-num** and **set max-node-num** commands. After setting up DECnet routing, check connectivity to hosts and other routers using the **show** and **stats** commands. After configuring DECnet, it is recommended that the configuration be saved using the **savecfg** <*file-name*> command. See the *PowerHub Hardware Reference Manual.*

22.1 Accessing the DECnet Subsystem

To access the dec subsystem, issue the following command from the runtime command prompt:

```
dec
```

Most of the commands in this chapter assume that the focus of the command prompt has been changed to “dec.” A few commands, such as **getmem**, are not in the dec subsystem. This chapter identifies such commands by listing their subsystem name with the command (ex: **atalk getmem**.).

22.1.1 Allocating Memory

Before using the dec subsystem, allocate memory for the subsystem by issuing the **getmem** command, as shown in the following example:

```
1:PowerHub:dec# atalk getmem
Memory allocated for DEC routing.
2:PowerHub:dec#
```

If memory has been allocated for DECnet routing at the time the configuration is saved with a **savecfg** command, the corresponding **getmem** command is placed in the configuration file ahead of other DECnet configuration commands. Thus, the **getmem** command need only be entered when first configuring DECnet routing.

NOTE

FORE Systems recommends that memory be allocated for the DECnet subsystem immediately after booting the PowerHub to ensure that the memory requested is available. For more information, refer to the *PowerHub Hardware Reference Manual*.

Memory cannot be de-allocated. To free allocated memory, make sure the configuration file does not contain a **getmem** command, then reboot the software.

Verify that memory has been allocated using the **rs** command. If memory has not been allocated, the command is not allowed to execute.

```
3:PowerHub:dec# rs
DECnet routing status:

Node/Segment            Management State   Routing State
---------               ----------------   -------------
DECnet-Forwarding       Disabled           Down
Segment  1.2            Disabled           Down
Segment  2.1            Disabled           Down
<additional rows omitted for brevity>
4:PowerHub:dec#
```

22.1.2 Node Configuration

When placed in a DECnet internetwork, the PowerHub acts as a standard router, capable of connecting many different DECnet networks together. It determines the identity and location of its neighbors through standard DECnet Phase IV protocols, and finds the closest path to each. It then uses this information to route packets that arrive at the input segments.

The DECnet Phase IV routing protocol calls for each node to have an "area number" (between 1 and 63), as well as a node ID (from 1 to 1023). For Level-1 networks (consisting only of Level-1 endnodes and routers), the area numbers are identical and unused. In Level-2 networks, an "area" is defined as a collection of several nodes with identical area numbers. These areas are connected by Level-2 routers. If the PowerHub is configured as a Level-2 router (with the **`set node-type area-router`** command), it uses an extended set of routing protocols that can connect nodes from different areas. Normal nodes can only route packets directly to other nodes within their area (those with matching area numbers). If they are called upon to send a packet to a node in another area, they send it to the "nearest" Level-2 router. This Level-2 router keeps track of routes to all other Level-2 routers, as well as routes to normal nodes within its area. This two-level hierarchy allows for a larger network with manageable routing tables.

When the PowerHub is configured as a Level-2 router, it locates all nodes in its area *and* all other Level-2 routers. Note that this places some restrictions on the topology of the network, described below. The PowerHub announces itself as a Level-2 router to the normal nodes in its area so that all inter-area packets are sent through it (thus a pair of Level-2 routers are needed for inter-area packets: one in each area). If the PowerHub is placed into a network that uses Level-2 routing, and the PowerHub is to serve as a Level-2 router, be sure to turn this option on (with **`set node-type area-router`**). If the PowerHub is not to be a Level-2 router, or if the network uses only Level-1 routing, turn this option off (with **`set node-type router`**). The default is to use only Level-1 routing.

The use of areas in DECnet Level-2 routing places some restrictions upon the topology of these networks:

- Each node must be able to get to each other node in its area without the use of Level-2 routers and without leaving its area. Consequently, all the nodes in a given area must form a contiguous group. If all nodes from other areas are removed, leaving only the nodes from this area, there can be no isolated nodes remaining. This restriction also applies to Level-2 router nodes.
- The set of all Level-2 router nodes must form a contiguous group so that any packet going from one Level-2 router to another can travel only through other Level-2 router nodes.
- There can not be multiple links between adjacent routers. If two routers are directly connected by more than one segment, the DECnet protocol must be enabled and running on only one of those links. Failure to ensure DECnet is running on only one link results in changes to the routing table every time the dou-

bly-connected nodes discover each other. Such a double connection causes the routing table to be continually flushed, resulting in poor performance and unreachable nodes.

This situation is represented graphically in Figure 22.1:

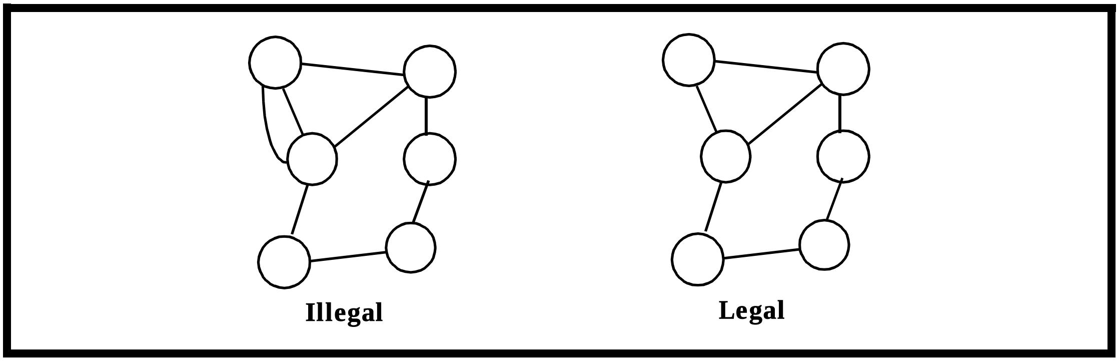


Figure 22.1 - Illegal Double Links

There is also a topological consideration that improves the efficiency of DECnet Level-2 networks. When a heavily populated broadcast medium is used, such as an Ethernet segment with several nodes, all the nodes on the same segment should be assigned the same area number. The reason is that two nodes with different area numbers must use Level-2 routing to communicate. Therefore this cable segment must have a pair of Level-2 routers on it (one for each area), and the communication path requires three hops, even though the nodes are on the same segment and could communicate directly by other protocols. To avoid these extra hops, all nodes that can communicate directly with each other should be placed in the same area by giving them identical area numbers.

22.1.3 DECnet Network Topology Restrictions

- All nodes in a given area must be connected.
- All Level-2 nodes must be connected.
- No redundant paths are allowed between adjacent routers.

Note that these restrictions do not prevent the same network segment from serving both Level-1 nodes and Level-2 nodes. Thus the same Ethernet segment can serve to connect Level-1 routers, Level-1 endnodes, and Level-2 routers. The requirement is that all of an area be contiguous; nodes from different areas can be on the same segment as long as data moving within one area does not have to pass through the other area's nodes in order to reach its destination.

22.1.4 Configuring the PowerHub as a DECnet Node

First, set the maximum node number used in this area. To do this, use the **max-node-num** parameter:

```
set max-node-num|mnn <value>
```

This determines the number of nodes that can exist within the PowerHub. The routing software ignores any packets from nodes outside this range. The default is 255, it must be increased to accommodate nodes with larger numbers. The DECnet protocol requires node numbers to be in the range 1 to 1023, the **max-node-num** parameter cannot be raised above 1023.

```
171:PowerHub:dec# set mnn 1023
Okay
```

Next, assign the PowerHub node ID. Use the **node-id** parameter:

```
set node-id|nid <area>.<node>
```

This command instructs the PowerHub to use the specified address for all DECnet communications. The <*area*> parameter must match the area in which the PowerHub has been placed; recall that the DECnet definition of “area” is the set of nodes that have the same area numbers. The <*node*> parameter can be any value that is unique among all nodes in the specified area.

```
172:PowerHub:dec# set nid 5.1023
Okay
```

Select the type of routing that needs to be done by this router. Use the **node-type** parameter command:

```
set node-type|ntp router|rt | area-router|a
```

This command determines what kind of routing the PowerHub performs. If “**router**” is chosen,the PowerHub performs only Level-1 routing. A Level-1 router keeps track of nodes within its own area only, and does not try to determine routes to other areas. If it receives data for another area, it sends it to the nearest Level-2 router. The PowerHub acts as a Level-1 router by default.

If “**area-router**” is chosen, the PowerHub also performs Level-2 routing. Level-2 is a superset of Level-1: the node routes data to nodes within its area, as well as find routes to other areas. All Level-2 routers find all other Level-2 routers (including those in other areas), and inter-area traffic is sent to a distant Level-2 router for local distribution.

By default, the router performs only Level-1 routing. No changes need to be made to this parameter if the PowerHub is going to be used as a Level-1 router. For Level-2 routing, enter: **set node-type area-router**.

```
173:PowerHub:dec# set nt area-router
Okay
```

Activate DECnet routing with the **enable dec** command:

enable dec

This is the primary command which turns on all of the DECnet routing software. However, to have a useful configuration, specify two or more segments that use DECnet. This is accomplished with the **penable dec** *<seglist>* command, as described in.

```
191:PowerHub:dec# penable dec
Okay
```

Verify the node configuration with the **show node-type** command.

```
195:PowerHub:dec# show node-type
DECnet node configuration
-------------------------
DEC-forwarding:     Enabled
Max-Area-Num:       63
Max-Node-Num:       1023
Max-Adj-Endnodes:   1023
Max-Adj-Routers:    128
Max-Cost-To-Area:   100
Max-Hops-To-Area:   16
Max-Cost-To-Node:   125
Max-Hops-To-Node:   30
Max-Visits:         60
Node-Type:          Area Rtr
Node-ID:            5.1023
Routing-State:      Up
Update-Time:        60 seconds
```

Now configure one or more segments to use DECnet forwarding.

22.1.4.1 Additional Node Commands

The following additional node commands are available to set various node parameters.

max-adj-endnodes|mae set *<value>*
Sets the number of endnode adjacencies supported by this router. The range for *<value>* is **1 - 1023**.

max-adj-endnodes|mae [show]
Display the results of setting the **max-adj-endnodes|mae** command.

max-adj-routers|mar set *<value>*
Sets the number of broadcast router adjacencies supported by this router. The range for *<value>* is **1 - 560**.

max-adj-routers|mar [show]
Display the results of setting the **max-adj-routers|mar** command.

max-area-num|man set *<value>*

Sets the maximum area number allowed in the entire network. The range for *<value>* is **1 - 63**. *<value>* must be greater than or equal to the maximum area in use.

max-area-num|man [show]

Display the results of setting this command, issue the following command:

max-cost-to-area|mca set *<value>*

Sets the maximum cost possible in a path to a reachable area. The range for *<value>* is **1 - 1022**. *<value>* must be greater than or equal to actual max hops to an area * **25**.

max-cost-to-area|mca [show]

Display the results of setting the **max-cost-to-area|mca set *<value>*** command.

max-cost-to-node|mcn set *<value>*

Sets the maximum cost possible in a path to a reachable node. The range for *<value>* is **1 - 1022**. *<value>* must be greater than or equal to actual max hops in area * **25**.

max-cost-to-node|mcn [show]

Display the results of setting the **max-cost-to-node|mcn set *<value>*** command.

max-hops-to-area|mha set *<value>*

Sets the maximum hops possible in a path to a reachable area. The range for *<value>* is **1 - 30**. *<value>* must be greater than or equal to actual max hops to any area.

max-hops-to-area|mha [show]

Display the results of setting the **max-hops-to-area|mha set *<value>*** command.

max-hops-to-node|mhn set *<value>*

Sets the maximum hops possible in a path to a reachable node. The range for *<value>* is **1 - 30**. *<value>* must be greater than or equal to actual max hops in an area.

max-hops-to-node|mhn [show]

Display the results of setting the **max-hops-to-node|mhn set *<value>*** command.

max-node-num|mnn set *<value>*

Sets the maximum node number allowed within this area. The range for *<value>* is **1 - 1023**. *<value>* must be greater than or equal to maximum node number in use.

max-node-num|mnn [show]

Display the results of setting the **max-node-num|mnn set *<value>*** command.

max-routers|mr pset *<value> <seglist>*

Sets the number of broadcast router adjacencies supported on the port(s) in *<seglist>*. *<seglist>* is a comma-separated list of ports or **all**. The range for *<value>* is **1 - 20**.

max-routers|mr [show] [*<seglist>*]

To display the results of setting this command, issue the following command:

hello-time|ht pset *<value> <seglist>*

Sets the interval for sending hello packets on the port(s) in *<seglist>*. *<seglist>* is a comma-separated list of ports or **all**. The range for *<value>* is **1 - 8191**

hello-time|ht [show] [*<seglist>*]

Display the results of setting the**hello-time|ht pset *<value> <seglist>*** command.

cost|c pset *<value> <seglist>*

Sets the cost for the ports in <seglist>. <seglist> is a comma-separated list of ports or all. The range for *<value>* **is 1 - 127**

cost [show] [*<seglist>*]

Display the results of setting the **cost|c pset *<value> <seglist>*** command.

max-visits set *<value> <seglist>*

Sets the maximum visits for a packet before the router assumes that the packet is looping. The range for *<value>* **is maxpath - 60.** *<value>* **must be greater than equal to the actual maximum path in the entire network**.

max-visits [show]

Display the results of setting the **max-visits set *<value> <seglist>*** command.

priority|pri pset *<value> <seglist>*

Sets the priority for the port(s) in *<seglist>*. *<seglist>* **is a comma-separated list of ports or all. The range for <value> is 0 - 127.**

priority|pri [show] *<seglist>*

Displays the results of setting the **priority|pre pset** command.

update-time|ut set *<secs>*

Sets background timer for sending routing updates. The range for *<secs>* **is 1 - 1200.**

update-time|ut [show]

Displays the results of setting the **update-time|ut set** command.

22.2 Segment Configuration

Once the PowerHub is configured to forward DECnet packets, designate one or more segments as DECnet segments to make the software interpret and forward the correct packets. This step also causes the software to transmit and accept routing control packets over these segments, enabling it to discover neighboring endnodes and routers. There are also several parameters associated with each segment that can be set to tune network performance.

22.2.1 Configuration

From the `dec` subsystem prompt, the only necessary segment configuration step is to enable DECnet forwarding for all segments attached to DECnet networks. The **`penable dec`**<*seglist*> command tells the software that DECnet packets may arrive over these segments and that they should be used for routing purposes:

`penable dec` ***`<seglist>`***

This command can be used to either enable or disable DECnet forwarding for each segment. The command uses the normal <*seg-ist*> syntax, which is a hyphen- and comma- separated list of segment numbers. For example, if segments 1, 2, and 3 are to be on DECnet networks, the command is:

`penable dec 2.1-2.4`

```
193:PowerHub:dec# penable dec 2.1-2.4
Port 2.1: Okay
Port 2.2: Okay
Port 2.3: Okay
Port 2.4: Okay
```

After enabling the segments, you can verify the segment configuration with the **`show priority`** command:

`show priority|pri [`*`<seglist>`*`]`

For example:

```
196:PowerHub:dec# dpp 1-2
DECnet port configuration (Port 1)
-----------------------------------
block-size:        1498
cost:              10
curr-adj-routers: 0
designated-rtr:    aa-00-04-00-1e-8a   (5.1023)
hello-time:        15 seconds
last-hello-sent:   12 seconds ago
mgmt-state:        Enabled
max-routers:       10
priority:          0
run-state:         Up
type:              Ethernet

DECnet port configuration (Port 2)
-----------------------------------
block-size:        1498
cost:              10
curr-adj-routers: 0
designated-rtr:    aa-00-04-00-1e-8a   (5.1023)
hello-time:        15 seconds
last-hello-sent:   12 seconds ago
mgmt-state:        Enabled
max-routers:       10
priority:          0
run-state:         Up
type:              Ethernet
```

At this point, verify the routing status of the DECnet software through the **`show routing-status`** command. This command shows the state of the global DEC forwarding as well as whether or not each segment is configured to route DECnet packets.

```
198:PH-4:dec# show rs
DECnet routing status:

Node/Port               Management State   Routing State
---------               ----------------   -------------
DEC-Forwarding          Enabled            Up

Port  1                 Enabled            Up
Port  2                 Enabled            Up
Port  3                 Enabled            Up
Port  4                 Disabled           Down
Port  5                 Disabled           Down
<remaining rows omitted for brevity>
```

In this listing, the Management State column refers to DECnet forwarding being enabled or disabled on each segment, while the Routing State column refers to the low-level hardware status. If that segment does not have a cable attached to it (and automatic segment-state detection is enabled), or if the segment has been disabled in the bridging subsystem, the Routing State shows “DOWN” instead of “UP.”

22.3 Display Commands

Using the display commands to:

- Look for adjacent DECnet routers in the network.
- Look for all DECnet endnodes adjacent to the PowerHub.
- Look at the DECnet routing table to verify that all the routes are present.

22.3.1 Verification of Routing

After the node and segments are configured, the PowerHub begins forwarding packets among nearby nodes. To verify that the PowerHub has identified its neighbors, use one of several display commands to examine routing tables and node lists. Figure 22.2 shows a sample DECnet network. The display commands shown give information about this configuration. Note that the PowerHub defined as node 5.1023, located on the left, is the one being monitored. It is serving as a Level-2 router for area 5, which consists of 7 nodes: itself, 5.34, 5.477, 5.45, 5.103, 5.553, and 5.811. There are 4 other areas, 37, 2, 8, and 59. In Figure 22.2, “endnodes” are depicted as a single circle. (Endnodes, such as non-routing workstations, are nodes not capable of forwarding packets.) Level-1 router nodes are shown as lightly-shaded rectangles.

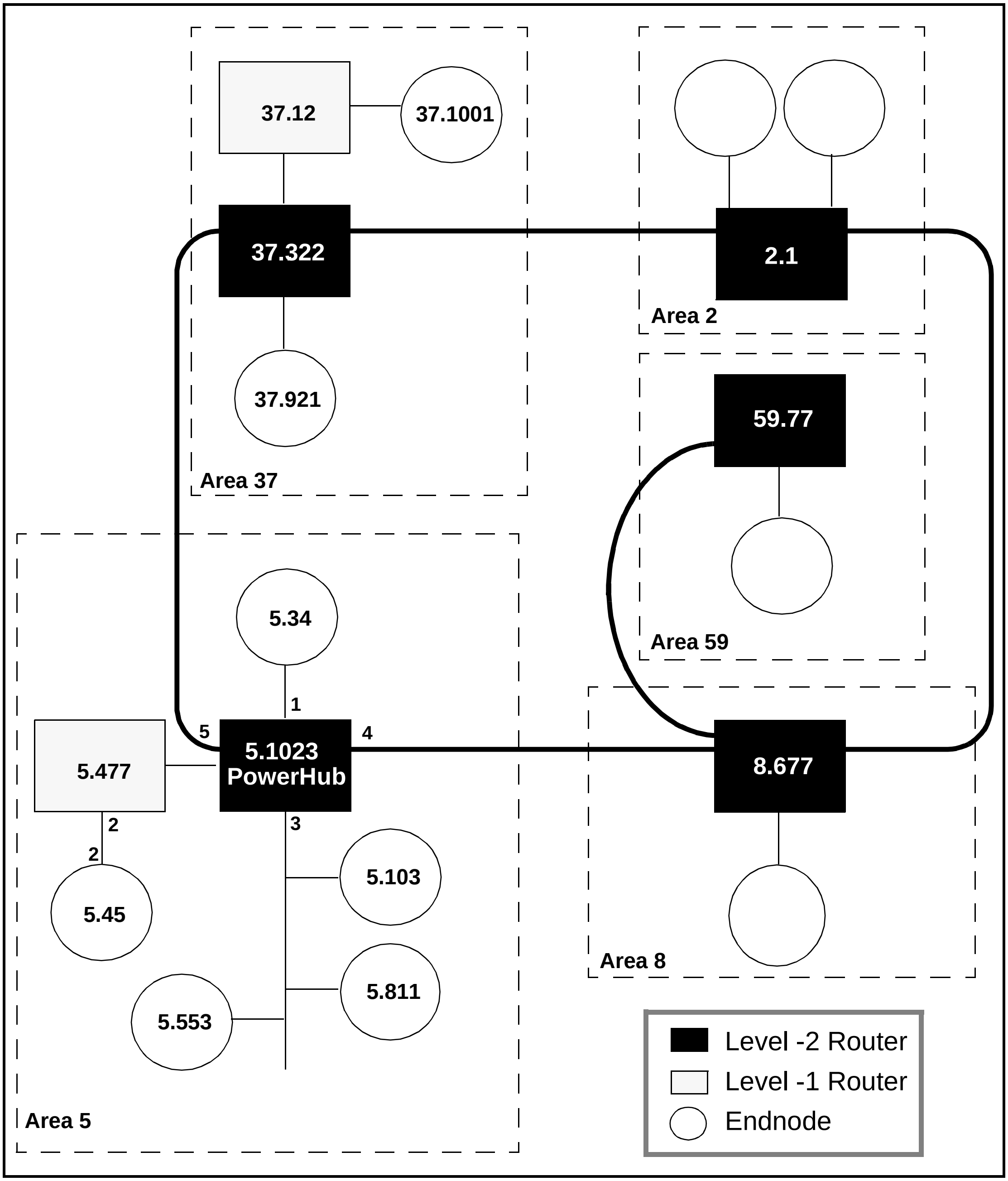


Figure 22.2 - Routing Verification

Note that nodes that are capable of routing, but appear on the periphery of networks (thus giving them nothing to route to), still qualify as routers and appear on the PowerHub listings as “routers” rather than “endnodes.” Level-2 router nodes are rectangles, and are connected with bold lines. All connections to the PowerHub are made through the segment numbers listed (1 through 5) by the small digits near the connecting lines. Also note that, while no endnodes are shown on the bold connections (links between departments, for example) between Level-2 routers, the protocols permit them to be there. For example, on the connection between 5.1023 and 37.322, endnodes or Level-1 routers for areas 5 and 37 could be attached. Each Level-2 router would recognize the nodes that belong to its area and forward packets to them.

22.3.2 Setting and Displaying Block-Size

The block-size command controls the size of internal routing tables. If it is a large network, block-size may need to be raised, but otherwise block-size should be kept low to conserve memory. To set the block-size of the internal routing tables, issue the following command:

```
block-size|bs pset <value> <seglist>
```

To display current block-size, issue the following command:

```
block-size|bs [show] [<seglist>]
```

22.3.3 Displaying Adjacent Routers

Look for adjacent routers in the network by typing:

```
adj[acent] [show] r[outer[s]]  [[a[ddr]=]<node>]
```

```
407:PowerHub:dec# show adj r
DECnet router adjacency table:
Adj Node ID Type      State  Seg   Blk Siz  Hel.Tim Prior. Age
--- ------- --------- -----  ----  -------  ------- ------ ---
1   5.477   Router     Up    2.1    1498      15      0     3
2   37.322  Area Rtr   Up    2.3    1498      15      0     3
3   8.677   Area Rtr   Up    2.5    1498      15      0     3
```

This command shows all the “adjacent” routers. In DECnet terminology, “adjacent” means “directly connected.” Thus nodes on the other end of 10Base-T links, or other sites on an Ethernet cable, are considered “adjacent.” A “router” is any node which can forward packets. Thus, this command shows all the directly connected routing nodes that the PowerHub has discovered. One is inside the PowerHub own area (5.477) and the other two are Level-2 routers (“Area Rtr”) in areas 37 and 8.

22.3.4 Displaying Adjacent Endnodes

Look for all endnodes adjacent to this router by entering:

adj[acent] [show] [end]node[s] [[a[ddr]=]<node>]

```
408:PowerHub:dec# show adj node
DECnet end-node adjacency table:
Adj   Node ID Type     State Seg  Blk Siz Hel.Tim  Prior. Age
----  ------- -------- ----- ---- ------- -------  ------ ----
1     5.34    End Node  Up   2.1  1498    10        0     9
2     5.811   End Node  Up   2.2  1498    10        0     9
3     5.103   End Node  Up   2.3  1498    10        0     9
4     5.553   End Node  Up   2.4  1498    10        0     9
```

This command shows all directly connected nodes that are “endnodes,” that is, those which cannot forward packets. The three nodes on the Ethernet cable are endnodes, as is node 5.34 (directly connected to segment 1). Note that node 5.45 is not adjacent, because this PowerHub cannot reach it directly.

22.3.5 Displaying the Route Table

Look at the routing tables to verify that all the routes are present by issuing the **display-route-tbl** command. Here is the syntax for this command:

show route|rt [*<disprestrict>*]

This command displays the route table, which is maintained by the DECnet routing software. It contains all the routes to nodes in this area that the PowerHub has found dynamically (DECnet does not provide for static, user-specified routes).

Here is an example of the display produced by this command:

```
409:PowerHub:dec# show rt
DECnet routing table:
Node       Seg    Next Hop             Hops  Cost
---------  -----  -----------------    ----  ----
area-rtr   -----  This-Rtr
5.34       1.1    ------                1    10
5.45       1.2    5.477                 2    10
5.103      1.3    ------                1    10
5.477      1.2    ------                1    10
5.553      1.3    ------                1    10
5.811      1.3    ------                1    10
5.1023     Local
```

Each entry contains the following information:

Node The address of the destination node.

Port The PowerHub segment that a packet destined for this node should leave on.

Next Hop The address of the next node a packet must pass through.

Hops The number of nodes the packet must pass through.

Cost A number reflecting the desirability of using this route.

From this table, it can be seen that Area 5 consists of 7 nodes: 34, 45, 103, 477, 553, 811, and 1023. The PowerHub is node 1023. The nodes on the Ethernet cable are 103, 553, and 811; they are accessible directly through segment 1.3. Two other nodes, 34 and 477, can be contacted directly through segments 1.1 and 1.2. One node, number 45, can only be reached through node 477, which is a router. Therefore the routing table shows that to send packets to node 45, the “Next Hop” is node 477, and that the node is two hops away from this one.

Note that the PowerHub, like all DECnet nodes, keeps track of the nearest Level-2 router. Since the PowerHub is configured as an area-router (Level-2), the nearest Level-2 router is itself. Consequently, the next hop listed for the “area-rtr” node (the one responsible for all inter-area routing) says “This-Rtr.”

If this router is configured as an area router (Level-2), look at the area table. This is a list of all known areas, along with the best way to get to them. To display this table, issue the following command:

```
area [show] [<area>]
```

```
410:PowerHub:dec# show area
DECnet area table:
Area  Port   Next Hop           Hops  Cost
----  -----  -----------------  ----  ----
   2    1.4  8.677                 2    20
   5  Local
   8    1.4  8.677                 1    10
  37    1.5  37.322                1    10
  59    1.4  8.677                 2    20
```

From this example, we can tell that the PowerHub is in area 5, and three other areas are accessible through the Level-2 router at 8.677, which is attached to the network on Segment 1.4. The other area is Area 37, available through segment 1.5.

If the PowerHub is configured as a Level-1 router (in a multi-area network), the “area-rtr” entry points to another node. As an example, imagine that the other router (node 5.477) is also a PowerHub.

To examine the route table on that hypothetical node, something like the following is displayed:

```
1:OtherPowerHub:dec# show area
DECnet routing table:
Node       Segment   Next Hop           Hops  Cost
---------  -----    -----------------  ----  ----
area-rtr     1.2    5.1023               1    10
5.45         1.1    5.45                 1    10
5.103        1.2    5.1023               2    10
5.477      Local
5.1023       1.2    5.1023               1    10
```

Examine the node and segment statistics to verify that the PowerHub is receiving data and control packets correctly.

22.3.6 Displaying Statistics

There are two types of statistics collected in the DECnet subsystem: node statistics and segment statistics. The node statistics are displayed with the **`stats show`** command, and contain information that is not associated with any particular segment. All the numbers displayed relate to errors or dropped packets, so the ideal display is all zeros. Here is an example of the **`stats show`** command:

```
411:PowerHub:dec# stats show n
DECnet node statistics (count since last stats clear):
node unreachable pkt loss    0
aged packet loss             0
node out-of-range pkt loss   0
oversized pkt loss           0
pkt format error             0
partial routing update loss  0
verification reject          0
routing table corrupted      0
no timers for updates        0
no bufs for sending hello    0
invalid hello from router    0
invalid hello from endnode   0
no room for router adj       0
no room for endnode adj      0
low priority rtr bumped      0
no bufs for lvl 1 update     0
lvl 1 msg format error       0
lvl 1 msg checksum error     0
lvl 1 msg area num error     0
no bufs for lvl 2 update     0
lvl 2 msg format error       0
lvl 2 msg checksum error     0
router moved to diff. port   0
end node moved to diff. port 0
```

As in other PowerHub subsystems, the DECnet software maintains two copies of the node statistics:

- Count since the last clear.
- Count since the last system reset.

Both counters increment when errors occur, but the **stats clear** command clears only the count since last clear. To display the count since the last reset, use the **-t** option with the **stats show** command, as shown in the following example. In this particular example, the Powerhub has just been rebooted and no statistics have yet been collected.

```
412:PowerHub:dec# stats show -t

DECnet node statistics (count since last stats reset):
node unreachable pkt loss    0
aged packet loss             0
node out-of-range pkt loss   0
oversized pkt loss           0
pkt format error             0
partial routing update loss  0
verification reject          0
routing table corrupted      0
no timers for updates        0
no bufs for sending hello    0
invalid hello from router    0
invalid hello from endnode   0
no room for router adj       0
no room for endnode adj      0
low priority rtr bumped      0
no bufs for lvl 1 update     0
lvl 1 msg format error       0
lvl 1 msg checksum error     0
lvl 1 msg area num error     0
no bufs for lvl 2 update     0
lvl 2 msg format error       0
lvl 2 msg checksum error     0
router moved to diff. port   0
endnode moved to diff. port  0
```

Segment statistics are collected in the same manner. These statistics are primarily counts of how many DECnet packets are routed through each segment. This can give an idea of where the most traffic is coming from, and may provide insight on how to better structure the network.

22.3.6.1 Displaying the Route Cache

To display the DECnet route cache, issue the following command:

```
cache [show] [<disprestrict>]
```

To clear the DECnet route cache, issue the following command:

```
cache clear
```

APPENDIX A Old/New User Interface Commands

The purpose of this appendix is to map the old user interface subsystem commands to the new user interface subsystem commands. New and old user interface commands are listed by subsystem. Those subsystems affected by the change are presented below.

Table A.1 - Global Commands

Old User Interface Command	New User Interface Command
alias [<*name*> **[**<*command*>**]]** With no arguments, shows all aliases. With one argument, shows the alias defined for <*name*>. With more than one argument, defines <*name*> as <*command*>.	**alias [**<*name*> **[**<*command*>**]]** With no arguments, shows all aliases. With one argument, shows the alias defined for <*name*>. With more than one argument, defines <*name*> as <*command*>.
mgmt showfile Displays a file.	**cat** <*filename*> Prints a floppy file to the tty.
help |? <*word*> [<*word*>...] Gets help for a command.	help |? <*word*> [<*word*>...] Gets help for a command.
history | hi Shows command history.	history | hi Shows command history.
histchars [<*ch1*>[<*ch2*>]] With no argument, shows current history characters. With an argument consisting of one or two characters, sets history characters to corresponding character values.	histchars [<*ch1*>[<*ch2*>]] With no argument, shows current history characters. With an argument consisting of one or two characters, sets history characters to corresponding character values.
logout or **bye** Ends the session.	logout | bye Ends the current UI session.
mgmt dir or **mgmt ls** Displays a directory.	ls | dir [<*filespec*>] Shows a directory listing for the floppy disk.
...no available help	mv | rename <*file1*> <*file2*> Changes the name of <*file1*> to <*file2*>

Table A.1 - Global Commands

Old User Interface Command	New User Interface Command
...no available help	set pnm multi|old [show] pnm Port Number Mode: **`multi`** Multi-part port numbers (<*slot*>.<*seg*>) **`old`** Old-style (<*vport*>)
...no available help	rcprompt enable Enables printing of command return codes in the next UI prompt. This feature is intended primarily for automated interactions with the PowerHub command-line interface.
...no available help	rcprompt disable Disables printing of command return codes in the next UI prompt.
`main readenv` Allows you to load environment files.	**`r[ea]denv`** <*file*> Executes environment file <file> in the context of the current UI session.
`mgmt remove` and **`mgmt fremove`** Deletes a file.	rm [-i] [-f] <*filespec*> Deletes files from the floppy disk.
`main saveenv` Allows you to save environment files.	saveenv <*file*> Saves the current UI environment to floppy file <*file*>.
`mgmt reboot` Reboots the system.	**`system reboot`** Reboots the system (cold start).
`mgmt tty2` Connects a management terminal to a TTY port on the PowerHub Packet Engine.	**`system enable|disable tty2`** Connects or disables a management terminal to a TTY port on the PowerHub Packet Engine.

Table A.1 - Global Commands

Old User Interface Command	New User Interface Command
main stty Controls scroll.	stty [-d[efault]] [-t <*tty*>] [rows <*#*>] [[-|+]more] [[-|+]dcd] [<*speed*>] [erase <*c*>] [kill <*c*>] [werase <*c*>] [intr <*c*>] [rprnt <*c*>] [stop <*c*>] [start <*c*>] Controls tty driver characteristics. With no parameters, shows the current tty settings.
main setuser Changes the management capability.	su [root|monitor] Changes userid to root or monitor (defaults to root).
subsystems|ss Lists the various subsystems.	subsystems|ss Lists the various subsystems.
telnet *<switch's IP address>* Initiates a TELNET session.	**telnet** *<switch's IP address>* Initiates a TELNET session.
main timedcmd Allows you to use timed commands.	**timedcmd|tc add** <*id*> <*time*> <*cmd*> Allows you to add timed commands.
main timedcmd Allows you to use timed commands.	**timedcmd|tc delete** <*id*> Allows you to delete timed commands.
main timedcmd Allows you to use timed commands.	**timedcmd|tc enable** <*id*> Allows you to enable timed commands.
main timedcmd Allows you to use timed commands.	**timedcmd|tc disable** <*id*> Allows you to disable timed commands.
tcpstack tcp-table and **tcpstack udp-table** Displays TELNET control characters.	**host status [show] tcp|udp** Displays current state of host systems.
tcpstack tcp-table and **tcpstack udp-table** Changes TELNET control characters.	**host status [show] tcp|udp** Changes TELNET control characters.
unalias Changes command alias.	unalias <*name*> Remove the alias definition for <*name*>.
oldui New command line interface.	zui New command line interface.

Table A.2 - System Commands

Old User Interface Command	New User Interface Command
`setbaud | sb` Sets the baud rate of the unit and saves it in NVRAM.	**`[set|show] baud [tty1|tty2] [1200|2400|4800|9600|19200]`** Sets the baud rate of the unit and saves it in NVRAM.
`bootinfo | bi` Prints boot information.	**`[show] bootinfo|bi`** Displays the boot log.
`card-swap | cs` card-swap | cs <card> in | out.	**`[enable|disable] card|card-swap|cs`** *<slot>* Enables or disables card *<slot>*. Logically inserts or removes card into system.
`card-swap | cs` card-swap | cs <card> in | out.	**`[show] card|card-swap|cs`** Displays current slot status.
`tcpstack showcfg` Displays the TCP connection.	**`config [show]`** Displays the Powerhub configuration.
`date | da` Shows the current system clock or sets the system clock.	**`[set|show] date [[YYMMDD]hhmm[.ss]]`** Shows the current system clock or sets the system clock.
...no available help	**`[show] dcd-detection|dcd]`** Displays status of dcd-detection.
`ethaddr | ea` Prints system ethernet address.	**`[show] ethaddr|ea`** Prints system ethernet address.
...no available help	**`[show] idprom|idp`** *<slot number>***`|all`** Display information in IDPROM of *<slot number>*.
`main passwd` Changes password for root or monitor (defaults to root)	**`passwd [root|monitor]`** Changes password for root or monitor (defaults to root).
`readcfg | rdcfg` Reads the configuration from the specified file or device.	**`readcfg|rdcfg [-v]`** *<file or device name>* Reads the configuration from the specified file or device.

Table A.2 - System Commands

Old User Interface Command	New User Interface Command
reboot | reboot Reboots the system (cold start).	**reboot** Reboots the system (cold start).
savecfg | svcfg Saves the configuration to the specified file or device.	**savecfg|svcfg** *<file or device name***>** Saves the configuration to the specified file or device as in the following examples: **tftp:** //1.2.3.4/configfile (use tftp to host 1.2.3.4) cfg (to local file "cfg"). **FM:** saved (to local Flash Module file "saved").
syslocn | sl Displays system location. This is used by SNMP managers.	**[show] syslocn** *<location>* Displays system location. This is used by SNMP managers.
sysname | sn Displays the system name.	**[show] sysname** *<location>* Displays the system name.
temperature | temp Displays the reading(s), in Celsius, from the temperature sensor(s) on the card(s) in the specified slot(s).	**[show] temperature|temp** *<slot number>***|all** Displays the reading(s), in Celsius, from the temperature sensor(s) on the card(s) in the specified slot(s).
tty2open | t2o Opens or closes tty2.	**enable|disable tty2** Opens or closes tty2.
...no available help	**uptime [show]** Shows elapsed time since last reboot.
version | ver Displays the software version string and **idprom** info of the card in the specified slot or all the cards in the PowerHub switch.	**version [show] [***<slot-number>***|all]** Displays the software version string and **idprom** info of the card in the specified slot or all the cards in the PowerHub switch.

Table A.3 - Media Commands

Old User Interface Command	New User Interface Command
showcfg | scf Show bridging-related configuration information.	**config [show] [***<params>***] [***<disp-restrictors>***]** Show bridging-related configuration information.
led-config | lc Displays the current led mode for all 13x1, 16x1, and Fast Ethernet cards in the system or in the specified slot.	**isstats [show] [***<params>***] [***<disp-restrictors>***]** Displays the current led mode for all 13x1, 16x1, and Fast Ethernet cards in the system or in the specified slot.
led-config | lc Displays the current led mode for all 13x1, 16x1, and Fast Ethernet cards in the system or in the specified slot.	**isstats clear|enable|disable** Clears, enables, or disables Inter-Segment Statistics.
led-config | lc Displays the current led mode for all 13x1, 16x1, and Fast Ethernet cards in the system or in the specified slot.	**[show] ledmode|lm [***<slot>***]** Displays the current led mode for all 13x1, 16x1, and Fast Ethernet cards in the system or in the specified slot.
led-config | lc Displays the current led mode for all 13x1, 16x1, and Fast Ethernet cards in the system or in the specified slot.	**set ledmode|lm** *<slot>* **ca|xr** Sets the led mode for the 13x1, 16x1, or Fast Ethernet card in the specified slot.
...no available help	**monitor [set] [from** *<monitor-spec>***] [to** *<monitor-spec>***] on** *<seglist>* **Use lan-monitor feature to emit certain packets on other segments.**
...no available help	**monitor [show] [***<seglist>***]** Display the list of which segments are currently being monitored. Show only those monitors which emit packets on segments in *<seglist>*.
...no available help	**monitor clear** Turn off all monitoring.

Table A.3 - Media Commands

Old User Interface Command	New User Interface Command
operating-mode | om Shows the operating mode of an ethernet controller.	**[show] operating-mode|om [***<seg-list>***|all]** Shows the operating mode of an ethernet controller.
operating-mode | om Sets the operating mode of an ethernet controller.	**set operating-mode|om** *<seglist>***|all fdx|lbk|flbk|declbk|normal|hdx** Sets the operating mode of an ethernet controller.
port-monitor | pm Enable or disable the receivers on the given ports.	**portreceive|pr penable|pdisable** *<port-list>* Enable or disable the receivers on the given ports.
port-monitor | pm Displays port-by-port statistics.	**portstats [show] [***<display-restrictor>***]** Displays port-by-port statistics.
port-monitor | pm Enables or disables the collection of port-by-port statistics.	**portstats enable|disable** Enables or disables the collection of port-by-port statistics.
...no available help	**segment penable|pdisable** *<segment-list>* Enables or disables the transmission and reception of all packets on the given segments.
set-portname | spn Sets the Segment-Name for the given segments.	**sset [segment]name** *<name> <seglist>* Sets the Segment-Name for the given segments.
set-portname | spn Sets the Segment-Name for the given segments.	**[show] [segment]name [***<seglist>***]** Show the Segment-Name for the given segments.
autoportstate | aps Displays the status of automatic Segment-State Detection for the given segments.	**[show] ssd [***<seglist>***]** Displays the status of automatic Segment-State Detection for the given segments.
autoportstate | aps Enables or disables automatic Segment-State Detection for the given segments.	penable | pdisable ssd *[<seglist>*] Enables or disables automatic Segment-State Detection for the given segments.

Table A.3 - Media Commands

Old User Interface Command	New User Interface Command
...no available help	sset ssdt[hresh] | ssdthreshold <*value*> <*seglist*> Set the threshold used for SSD on the given segments.
...no available help	status [show] [<*params*>] [<*display-restrictors*>] Displays port-level status for UTP ports.
...no available help	stats [show] [-p | -s] [<*params*>] [<*display-restrictor*>] Displays media-level statistics.
...no available help	stats [clear] Clear media-level statistics.

Table A.4 - Host Commands

Old User Interface Command	New User Interface Command
...no available help	**config [show] tcp|fi[lters]|ru[les]|tem[plates] [**<*disprestrictions*>**]** Displays subsystem configuration.
...no available help	**filter define** <*filter-number*> <*template-number*>**[,**<*template-number*>...**]** Defines filter <*filter-number*> to be the ordered list of specified templates.
...no available help	**filter undefine** <*filter-number*> Deletes the definition for filter <*filter-number*>.
...no available help	**filter [show] [f[ilter]=**<*filter-number*>**] [**<*seglist*>**]** Shows the host filter rules associated with the specified segments.

Table A.4 - Host Commands

Old User Interface Command	New User Interface Command
...no available help	**`filter attach`** <*rule-number*> **`r[eceive]`** <*seglist*> Attaches rule <rule-number> to the receive packet stream on the specified segments. <*seglist*> may be "all".
...no available help	**`filter detach r[eceive]`** <*seglist*> Detaches the host filtering rule(s) from the receive or transmit packet streams on the specified segments. <*seglist*> may be "all".
...no available help	set kainterval | kai <*time*> Sets interval between keep-alive sends (Range: 30-240).
...no available help	set kadelay | kad <*time*> Sets time to elapse before sending keep-alives (Range: 5-30).
...no available help	kill <*connection-id*> Kills the TCP connection specified by the ID.
rate-set | rse <*slot*> <*rate-group*> <*rate*> Configures the specified slot's rate group to a given rate.	stats [show | clear] [-i] [-t] tcp | tel[net] | udp Displays or clear subsystem statistics.
rate-showcfg | rscf <*slot*> | all Shows rate group information of slot specified.	status [show] tcp | udp Displays current state of host systems.
atm-vc-show | avc <*segment*> | all Displays all VCs which are active on specified ATM segment(s).	template [show] *[<template-number>*] Displays template contents. If no <*template-number*> is specified, shows template contents for all templates.

Table A.4 - Host Commands

Old User Interface Command	New User Interface Command
bridge set|se template|t Defines a filter template with the specified parameters.	template define <*template-number*> [sipa=<*ipaddr*>] [sipm=<*ipaddr*>] [dipa=<*ipaddr*>] [dipm=<*ipaddr*>] [ipproto={tcp,udp}|<*protonum*>] [tsport{=|<|>}<*wks*>|<*portnum*>] [tdport{=|<|>}<*wks*>|<*portnum*>] [action=pass|block] Defines a filter template with the specified parameters. Ip addresses and masks may be entered in either hexadecimal or in dotted-quad notation. 'tsport' and 'tdport' are transport-protocol source and destination ports, and can only be used if 'ipproto' matches against TCP or UDP. The action parameter defaults to "block".
...no available help	template undefine <*template-number*> Deletes the definition for template <*num*>.

Table A.5 - NVRAM Commands

Old User Interface Command	New User Interface Command
set <*variable*> <*value*> Sets the boot order.	**set bo** <*value*> Sets the boot order.
show <*variable*> Displays the value of the boot order string.	**[show] bo** Displays the value of the boot order string.
unset <*variable*> Clears the boot order string (will default to floppy/flash boot).	**unset bo** Clears the boot order string (will default to floppy/flash boot).
show <*variable*> Displays the "myip" nvram variable.	**[show] myip** Displays the "myip" nvram variable.
set <*variable*> <*value*> Sets the "myip" nvram variable to the specified IP address value.	**set myip** <*ipaddr*> Sets the "myip" nvram variable to the specified IP address value.

Table A.5 - NVRAM Commands

Old User Interface Command	New User Interface Command
unset *<variable>* Unsets the value of the "myip" nvram variable.	**unset myip** Unsets the value of the "myip" nvram variable.
show *<variable>* Display the "mysm" nvram variable.	**[show] mysm** Display the "mysm" nvram variable.
set *<variable> <value>* Sets the "mysm" nvram variable to the specified IP address mask.	**set mysm** *<ipaddr-mask>* Sets the "mysm" nvram variable to the specified IP address mask.
unset *<variable>* Unsets the value of the "mysm" nvram variable.	**unset mysm** Unsets the value of the "mysm" nvram variable.
show *<variable>* File Server Address. Displays the "fsip" nvram variable.	**[show] fsip** File Server Address. Displays the "fsip" nvram variable.
set *<variable> <value>* File Server Address. Sets the "fsip" nvram variable to the specified IP address.	**set fsip** *<ipaddr>* File Server Address. Sets the "fsip" nvram variable to the specified IP address.
unset *<variable>* File Server Address. Unsets the value of the "fsip" nvram variable.	**unset fsip** File Server Address. Unsets the value of the "fsip" nvram variable.
show *<variable>* Boot Gateway Address. Displays the "gwip" nvram variable.	**[show] gwip** Boot Gateway Address. Displays the "gwip" nvram variable.
set *<variable> <value>* Boot Gateway Address. Sets the "gwip" nvram variable to the specified IP address.	**set gwip** *<ipaddr>* Boot Gateway Address. Sets the "gwip" nvram variable to the specified IP address.
unset *<variable>* Boot Gateway Address. Unsets the value of the "gwip" nvram variable.	**unset gwip** Boot Gateway Address. Unsets the value of the "gwip" nvram variable.
show *<variable>* Post-crash Behavior. Displays the value of the "crashreboot" nvram variable.	**[show] crashreboot** Post-crash Behavior. Displays the value of the "crashreboot" nvram variable.

Table A.5 - NVRAM Commands

Old User Interface Command	New User Interface Command
`set` *<variable> <value>* Post-crash Behavior. Sets the "crashreboot" nvram variable.	**`set crashreboot`** Post-crash Behavior. Sets the "crashreboot" nvram variable.
`unset` *<variable>* Post-crash Behavior. Unset the "crashreboot" nvram variable.	**`unset crashreboot`** Post-crash Behavior. Unset the "crashreboot" nvram variable.
`show` *<variable>* Displays the slot segment allocation. This form shows all slots.	**`[show] slotsegs`** Displays the slot segment allocation. This form shows all slots.
`show` *<variable>* Displays the slot segment allocation. This form shows the allocation for slot <*n*>.	**`[show] slotsegs[`***<n>***`]`** Displays the slot segment allocation. This form shows the allocation for slot <*n*>.
`set` *<variable> <value>* Set the slot segment allocation for slot <*n*>.	**`set slotsegs[`***<n>***`]`** *<segment-count>* Set the slot segment allocation for slot <*n*>.
`unset` *<variable>* Unset the slot segment allocation for slot <*n*>.	**`unset slotsegs[<n>]`** Unset the slot segment allocation for slot <*n*>.
...no available help	**`[show] md5key`** Displays the MD5 keys allocation. This form shows all keys; the second form shows the allocation for key <keyid>.
...no available help	**`[show] md5key[`***<keyid>***`]`** Displays the MD5 keys allocation. This form shows the allocation for key <keyid>.
...no available help	**`set md5key[`***<keyid>***`]`** *<key-string>* Sets the MD5 key allocation for the keyid.
...no available help	**`unset md5key[`***<keyid>***`]`** Unsets the MD5 key allocation for the keyid.

Table A.6 - TFTP Commands

Old User Interface Command	New User Interface Command
show <*variable*> Displays a TFTP variable.	**[show] server** Shows the IP address of the default tftp server host.
set <*variable*> <*value*> Sets, displays, and removes the default TDCTP Server.	**set server** <*ipaddr*> Sets the IP address of the default tftp server host.
unset <*variable*> Unsets a TFTP variable.	**unset server** Clears the IP address of the default tftp server host.
get [-h <*host*>**] [-a]** <*remfile*> **[**<*local-file*>**|tty]** Downloads or displays a file.	**get [-h** <*host*>**] [-a]** <*remote-file*> **[**<*local-file*>**|tty]** Transfers <*remote-file*> from host to disk file <*local-file*> (use remote filename as default) or to tty.
put [-h <*host*>**] [-a]** <*localfile*> **[**<*rem-file*>**]** Uploads a file.	**put [-h** <*host*>**] [-a]** <*localfile*> **[**<*remote-file*>**]** Transfers <*localfile*> from disk to host's <*remote-file*> (use local filename as default).
rdcfg [-h <*host*>**]** <*remfile*> Reads a configuration file.	**readcfg|rdcfg [-v] [-h** <*host*>**]** <*remote-file*> Executes configuration file as retrieved from host's <*remote-file*>.
svcfg [-h <*host*>**]** <*remfile*> Saves a configuration file.	**savecfg|svcfg [-h** <*host*>**]** <*remote-file*> Save configuration to host's <*remote-file*>

Table A.7 - Bridge Commands

Old User Interface Command	New User Interface Command
set | se Sets the bridge table aging delay to <*time*>.	**set aging** <*time*> Sets the bridge table aging delay to <*time*>.

Table A.7 - Bridge Commands

Old User Interface Command	New User Interface Command
set | se Disables aging.	**unset aging** Disables aging.
showcfg|scf Displays the current bridging status for all segments.	**bridging|br [show]** Displays the current bridging status for all segments.
bridging | br Enables bridging on the requested segments.	**bridging|br penable** *<segment-list>* Enables bridging on the requested segments.
bridging | br Disables bridging on the requested segments.	**bridging|br pdisable** *<segment-list>* Disables bridging on the requested segments.
bridge-table | bt Adds a permanent entry to the bridge table.	**bt add** *<ethaddr> <seglist>* Adds a permanent entry to the bridge table.
bridge-tableclear | btc Deletes a permanent entry from the bridge table.	**bt delete** *<ethaddr>* Deletes a permanent entry from the bridge table.
bridge-table | bt Shows the contents of the bridge table.	**bt [show] [-h] [-m] [-t]** *[<disprestrict>***]** Shows the contents of the bridge table.
bridge-tableclear | btc Clears the bridge table of learned entries.	**bt clear** Clears the bridge table of learned entries.
bridge display-cache Displays the bridge cache.	**cache [show] [***<disprestrict>***]** Shows the bridge cache.
flush-cache Flushes the bridge cache.	**cache clear** Clears the bridge cache.

Table A.7 - Bridge Commands

Old User Interface Command	New User Interface Command
showcfg | scf Shows the bridging subsystem configuration variables.	**config [show] [***<params>***] [***<disprestrict>***]** Display configuration of the bridge subsystem. The following parameters may be given to specify the type of information displayed: **vars** aging, local-filtering, ip-bridging **groups** bridging groups **templates** filter templates **rules** filter rules **filters** where filters are applied **st** spanning tree configuration
set | se Attaches a filter.	**filter attach** *<rnum>* **receive|transmit** *<seglist>* Attaches a filter. This form attaches a rule (defined with "lrule define") to the incoming or outgoing stream on a segment.
set | se Attaches a filter.	**filter attach** *<rnum>* **node** *<ethaddr>* This form applies a rule to packets coming from or going to the node *<ethaddr>*.
...no available help	**filter detach receive|transmit** *<seglist>* Remove a filter. This form removes a logical rule from the incoming or outgoing packet stream on a segment.

Table A.7 - Bridge Commands

Old User Interface Command	New User Interface Command
...no available help	**`filter detach`** *<rnum>* **`node`** *<ethaddr>* This form removes a logical rule's association with packets coming from or going to node *<ethaddr>*.
`getmem` Allocate memory for bridge table MIB processing.	**`getmem [br]mib`** Allocate memory for bridge table MIB processing.
`bridge set group` Adds a bridge (network) group.	**`pset group`** *<groupname> <seglist>* Groups *<seglist>* into a group called *<groupname>*.
`set group del` Deletes a bridge (network) group.	**`punset group`** *<groupname>* Deletes the group named *<groupname>*.
`ipx-br-translation | ibt` Displays the IPX network number(s) added for translation bridging between Ethernet and FDDI.	**`ipx-br-translation|ibt [show] [`***<network>***`]|[-t]`** Displays the IPX network number(s) added for translation bridging between Ethernet and FDDI.
`ipx-br-translation | ibt` Adds an IPX network for translation bridging between Ethernet and FDDI.	**`ipx-br-translation|ibt add`** *<network> <ethernet-encap> <fddi-encap>* Adds an IPX network for translation bridging between Ethernet and FDDI.
`ipx-br-translation | ibt` Deletes a previously added IPX network number or all networks.	**`ipx-br-translation|ibt delete`** *<network>***`|all`** Deletes a previously added IPX network number or all networks.
`ipx-br-translation | ibt` Enables IPX translation bridging.	**`ipx-br-translation|ibt enable`** Enables IPX translation bridging.
`ipx-br-translation | ibt` Disables IPX translation bridging.	**`ipx-br-translation|ibt disable`** Disables IPX translation bridging.
`br penable|pdisable` Enables or disables bridge learning on the specified segments.	**`penable learning`** *<seglist>* Enables bridge learning on the specified segments.

Table A.7 - Bridge Commands

Old User Interface Command	New User Interface Command
br penable|pdisable Enables or disables bridge learning on the specified segments.	**pdisable learning** *<seglist>* Disables bridge learning on the specified segments.
set | se Defines a logical filtering rule.	**lrule define** *<rnum>* *<rule-statement>* Defines a logical filtering rule.
set | se Undefines a logical filtering rule.	**lrule undefine** *<rnum>* Undefines a logical filtering rule.
bridge spantree enable|disable Enable or disable bridging on the requested port(s)	**enable|disable spantree|st** Enables or disables spanning tree.
bridge spantree enable|disable Enable or disable bridging on the requested port(s)	**enable|disable spantree|st fast-hello** Enables or disables spanning tree fast hellos.
bridge spantree enable|disable Enable or disable bridging on the requested port(s)	**spantree|st set maxage** *<time>* Sets the spanning tree maxage parameter to *<time>*.
bridge spantree enable|disable Enable or disable bridging on the requested port(s)	**spantree|st set hello** *<time>* Sets the spanning tree hello parameter to *<time>*.
bridge spantree enable|disable Enable or disable bridging on the requested port(s)	**spantree|st set fwddelay** *<time>* Sets the spanning tree fwddelay parameter to *<time>*.
bridge spantree enable|disable Enable or disable bridging on the requested port(s)	**spantree|st set fast-hello** *<time>* Sets the spanning tree fast-hello parameter to <time>.
bridge spantree enable|disable Enable or disable bridging on the requested port(s)	**spantree|st set high-util** *<percentage>* Sets the spanning tree high-util parameter to *<percentage>*.
bridge spantree enable|disable Enable or disable bridging on the requested port(s)	**spantree|st set low-util** *<percentage>* Sets the spanning tree low-util parameter to *<percentage>*.

Table A.7 - Bridge Commands

Old User Interface Command	New User Interface Command
`bridge spantree enable|disable` Enable or disable bridging on the requested port(s)	**`spantree|st set bridge-priority|bp`** *<priority>* Sets the spanning tree bridge-priority parameter to <priority>.
`bridge spantree enable|disable` Enable or disable bridging on the requested port(s)	**`spantree|st sset seg-priority|sp`** *<priority> <seglist>* Sets the seg-priority on the segments in *<seglist>* to *<priority>*.
`bridge spantree enable|disable` Enable or disable bridging on the requested port(s)	**`spantree|st sset path-cost|pc`** *<path-cost> <seglist>* Sets the path-cost on the segments in *<seglist>* to *<path-cost>*.
`stats-clear | sc` Clears statistics.	**`stats clear`** Clears statistics.
`stats | s` Shows statistics.	**`stats [show]`** Shows statistics.
`state` Show spanning tree status.	**`status [show]`** Show spanning tree status.
`set | se` Define a filter template.	**`template define`** *<tnum>* **`[size=w|h|b]`** **`off=`***<num> mask=<mask>* **`comp=`***<comp>* Define a filter template. The keyword parameters may be given in any order. *<mask>* is a 32-bit mask given in hex (eight hexadecimal digits). *<comp>* is a 32-bit value to compare, given in hex (eight hexadecimal digits).
`set | se` Undefine a filter template.	**`template undefine`** *<tnum>* Undefine a filter template.

Table A.8 - IP Subsystem Commands

Old User Inerface Command	New User Interface Command
add-arp | aa Enables or disables learning arp from incoming traffic. The default is to enable auto-learn.	**arp enable|disable auto-learn** Enables or disables learning arp from incoming traffic. The default is to enable auto-learn.
add-arp | aa Adds an arp table entry for <*ipaddr*> pointing towards <*ethaddr*> on <*seglist*>.	**arp add [-p]** <*ipaddr*> <*ethaddr*> <*seglist*> Adds an arp table entry for <*ipaddr*> pointing towards <*ethaddr*> on <*seglist*>.
arp-table | at Shows the ARP table.	**arp [show] [-r] [-t] [-s] [**<*disp-restrictors*>**]** Shows the ARP table.
del-arp | da Deletes an entry from the ARP table for the given IP address.	**arp delete** <*IPaddr*> Deletes an entry from the ARP table for the given IP address.
arp-tableclear | atc Clear all manually-added entries from the ARP table.	**arp clear** Clear all manually-added entries from the ARP table.
set-arpage | saa Show ARP aging time.	**arp [show] age** Show ARP aging time.
set-arpage | saa Set ARP aging time.	**arp set age** <*time*> Set ARP aging time (default 5 minutes, minimum 1 minute) <*time*> may be given as <*seconds*>, <*minutes*>:<*seconds*>, or <*hours*>:<*minutes*>:<*seconds*>.
set-arpage | saa off Turn off ARP aging.	**arp unset age** Turn off ARP aging.
set|se bridge-net-bcast|bnb enl|dis Enables or disables Bridge-Net-Broadcasts.	**enable|disable bridge-net-broadcast|bnb** Enables or disables Bridge-Net-Broadcasts.
display-routecache | dc Shows the IP router cache for the given segments.	**cache [show] [**<*seglist*>**]** Shows the IP router cache for the given segments.
flush-routecache | fc Flushes the IP router cache.	**cache clear** Flushes the IP router cache.

Table A.8 - IP Subsystem Commands

Old User Inerface Command	New User Interface Command
`showcfg | scf` Displays IP configuration.	**`config [show] ip`** Displays IP configuration.
`show-helper | sh` Displays helper configuration.	**`config [show] helper`** Displays helper configuration.
`ip-fil-acs-ctrl | ifa` Defines filter *t*o be the ordered list of specified templates.	**`filter define`** <*filter-number*> <*template-number*>**`[,`**<*template-number*>**`...]`** Defines filter <*filter-number*> to be the ordered list of specified templates.
`ip-fil-rule|ifr del|d` Deletes the definition for filter**.**	**`filter undefine`** <*filter-number*> Deletes the definition for filter <*filter-number*>.
`ip-fil-stats | ifs` Shows the ordered list of templates for one or all filters, or shows the filters associated with the specified (or all) segments.	**`filter [show] [f[ilter]=`**<*filter-number*>**`] [`**<*seglist*>**`]`** Shows the ordered list of templates for one or all filters, or shows the filters associated with the specified (or all) segments.
`ip-fil-acs-ctrl|ifa add|a` Attach filter <*filter-number*> to the receive or transmit packet stream on the specified segments. <*seglist*> may be "all".	**`filter attach`** <*filter number*> **`r[eceive]|t[ransmit]`** <*seglist*> Attach filter <*filter-number*> to the receive or transmit packet stream on the specified segments. <*seglist*> may be "all".
`ip-fil-acs-ctrl|ifa del|d` Deletes the attached filter.	**`filter detach r[eceive]|t[ransmit]`** <*seglist*> Deletes the attached filter.
`set|se fwd-pkts-with-srcrt-option|fps enl|dis` Enable or disable forwarding of IP packets containing either the Loose-source-route or the Strict-source-route option.	**`enable|disable fwd-pkts-with-srcrt-option|fps`** Enable or disable forwarding of IP packets containing either the Loose-source-route or the Strict-source-route option.
`add-helper-port | ahp` Adds a UDP Helper entry or more Default ports.	**`helper add`** <*IPaddr*> **`[`**<*UDPportlist*>**`]`** <*seglist*> Adds a UDP Helper entry or more Default ports.

Table A.8 - IP Subsystem Commands

Old User Inerface Command	New User Interface Command
add-helper-port | ahp Adds a UDP Helper entry or more Default ports.	**helper add -d** *<more UDP ports>* Adds a UDP Helper entry or more Default ports.
del-helper | dh Deletes a UDP Helper entry or default ports.	**helper delete** *<IPaddr> <UDP-portlist>>***|default[s]|all** *<seglist>* Deletes a UDP Helper entry or default ports.
del-helper | dh Deletes a UDP Helper entry or default ports.	**helper delete -d** *<UDP ports to remove>* Deletes a UDP Helper entry or default ports.
show-helper-port | shp Shows the UDP-helper table, or list current "default UDP ports" sorted in order by IP address.	**helper [show] [-p|-s]** Shows the UDP-helper table, or list current "default UDP ports" sorted in order by IP address.
show-helper-port | shp Shows the UDP-helper table, or list current "default UDP ports" sorted in order by IP address.	**helper [show] -d** Shows the UDP-helper table, or list current "default UDP ports" sorted in order by IP address.
show-helper | sh Displays helper configuration.	**config [show] helper** Displays helper configuration.
stats [show] [-t] helper Show helper statistics.	**stats [show] [-t] helper** Show helper statistics.
clear-helper-stats | chs Clear helper statistics.	**stats clear helper** Clear helper statistics.
showcfg | scf Displays IP configuration.	**config [show] ip** Displays IP configuration.
stats | s Shows IP statistics.	**stats [show] [-t] ip** Shows IP statistics.
stats-clear | sc Clears IP statistics.	**stats clear ip** Clears IP statistics.
set|se ipDefaultTTL|ittl Sets value of default TTL to use (range: 1 - 255).	**ipdefaultttl|ittl set** *<value>* Sets the default Time-To-Live for outgoing IP packets.

Table A.8 - IP Subsystem Commands

Old User Inerface Command	New User Interface Command
...no available help	**`ipdefaultttl|ittl [show]`** Displays the default Time-To-Live for outgoing IP packets.
`add-interface | ai` Adds an IP interface to the given segment(s).	**`it|interface add`** *<vlanid> <ipaddr>***`[/`***<prefixlen>***`|`***<mask>***`] [br[oadcast] 0|1] [met[ric]`** *<metric>***`]`** Adds an IP interface to the given segment(s). If <mask> is not specified then "natural" subnet mask (class A, B, or C address mask) is used.
`del-interface | d` Adds an IP interface to the given segment(s).	**`it|interface del[ete] [-p]`** *<vlanid>***`|all`** *<ipaddr>***`|all`** Deletes (possibly several) IP interfaces.
`iinterface-table | it` Shows the list of configured IP interfaces.	**`it|interface [show] [-s] [`***<disprestrictors>***`]`** Shows the list of configured IP interfaces.
...no available help	**`vlan add`** *<vlanid> <seglist>* Creates vlan *<vlanid>*.
...no available help	**`vlan del[ete]`** *<vlanid>* Deletes vlan *<vlanid>*.
...no available help	**`vlan [show]`** Shows the list of configured VLANs.
...no available help	**`vlan`** *<vlanid>* **`tset seglist`** *<seglist>* Changes parameters of vlan *<vlanid>*.
...no available help	**`vlan`** *<vlanid>* **`tset new-name`** *<new vlanid>* Changes parameters of vlan *<vlanid>*.
...no available help	**`load-balance|lb enable|disable`** Enable or disable load-balancing on equal-cost routes.

Table A.8 - IP Subsystem Commands

Old User Inerface Command	New User Interface Command
...no available help	**`loop-detection|ld set time`** *<value>* Sets the Time interval (in minutes) for sending out "loopback detection" packet for each configure IP interface. If not set, the default value will be 10 minutes.
...no available help	**`loop-detection|ld [show]`** Display the IP Loop Detection Table.
...no available help	**`loop-detection|ld enable|disable`** Enables or disables loop-detection. If enabled, a special "loopback detect" packet will be sent on each outbound segment that has at least one IP address. If disabled, no "loopback detect" packets will be sent.
`ping | pi` Do an ICMP PING to attempt to reach a given *<ipaddr>*.	**`ping [-t`** *<timeout>***`] [-size`** *<size>***`]`** *<ipaddr>* Do an ICMP PING to attempt to reach a given *<ipaddr>*.
`proxy-arp | pa` Show the status of proxy-ARP for the given segments.	**`proxy-arp [show] [`***<seglist>***`]`** Show the status of proxy-ARP for the given segments.
`proxy-arp | pa` Enables or disables proxy-ARP for the given segments.	**`proxy-arp penable|pdisable`** *<seglist>* Enables or disables proxy-ARP for the given segments.
...no available help	**`rdm [show]`** Shows RDM parameters.
...no available help	**`rdm nenable|ndisable`** *<ipaddr>* Enables or disables advertisement of the IP address. The default is to enable advertisement.

Table A.8 - IP Subsystem Commands

Old User Inerface Command	New User Interface Command
...no available help	**`rdm nset AdvertisementAddress multicast|broadcast`** *<ipaddr>* Either sends out advertise messages to the all-systems multicast address, 224.0.0.1. or to the limited-broadcast address, 255.255.255.255. The default is to use the all-systems multicast address.
...no available help	**`rdm nset preference`** *<preference> <ipaddr>* The Preference Level (integer) transmitted in the advertise messages. The default is zero.
...no available help	**`rdm nset interval`** *<time> <ipaddr>* The allowed values are between 0:04 and 30:00. The approximate interval between transmitting the advertise messages. The default time is 10:00.
`route-table | rt` Displays the requested IP multicast routing table (in column format).	**`route|rt [show] [-c|-r|-s|-o] [-d|-t] [-f] [-a] [`***<disprestrictors>***`]`** Displays the requested IP multicast routing table (in column format).
`add-route | ar` Adds a static route.	**`route|rt add [-s] [-d]`** *<destination> <gw-ipaddr>* **`metric`** *<metric>* **`segment`** *<segment>* Adds a static route.
`del-route | dr` Deletes a static route.	**`route|rt del[ete]`** *<destination> <gw-ipaddr>* Deletes a static route.
...no available help	**`route|rt enable`** *<destination> <gw-ipaddr>* Marks the route as UP, available for traffic.
...no available help	**`route|rt disable`** *<destination> <gw-ipaddr>* Marks the route as DOWN, not available for traffic.

Table A.8 - IP Subsystem Commands

Old User Inerface Command	New User Interface Command
`set rnb enl|dis` Enables or disables Route-Net-Broadcasts. If disabled, incoming IP datagrams with an IP broadcast address but MAC-layer unicast address (also known as "directed broadcasts") will be dropped. If enabled, they will be routed to local or distant IP subnets as necessary.	**`enable|disable route-net-broadcast|rnb`** Enables or disables Route-Net-Broadcasts. If disabled, incoming IP datagrams with an IP broadcast address but MAC-layer unicast address (also known as "directed broadcasts") will be dropped. If enabled, they will be routed to local or distant IP subnets as necessary.
`set sid enl|dis` Enables or disables sending of ICMP redirect messages.	**`enable|disable send-icmp-redirect|sir`** Enables or disables sending of ICMP redirect messages.
`stats | s` Show statistics for the given subsystem (or all subsystems).	**`stats [show] [-t] [-p] [arp|icmp|ip|helper]`** Show statistics for the given subsystem (or all subsystems).
`stats-clear | sc` Show statistics for the given subsystem (or all subsystems).	**`stats clear arp|icmp|ip|helper|all`** Clears statistics for the given subsystem (or all subsystems).
`ip-fil-template|ift show|s` Displays template contents.	**`template [show] [`**<*template-number*>**`]`** Displays template contents. If no <*template-number*> is specified, show template contents for all templates.

Table A.8 - IP Subsystem Commands

Old User Inerface Command	New User Interface Command
`ip-fil-template|ift chng|c` Define a filter template with the specified parameters. Ip addresses and masks may be entered in either hexadecimal or in dotted-quad notation. 'tsport' and 'tdport' are transport-protocol source and destination ports, and can only be used if 'ipproto' matches against TCP or UDP. The action parameter defaults to "block".	**`template define`** *<template-number>* **`[sipa=`***<ipaddr>* **`sipm=`***<ipaddr>***`]`** **`[dipa=`***<ipaddr>* **`dipm=`***<ipaddr>***`]`** **`[ipproto={tcp,udp}|`***<protonum>***`]`** **`[ipopt=srcrt]`** **`[tcptype=conreq]`** **`[tsport{=|<|>}`***<wks>***`|`***<portnum>***`]`** **`[tdport{=|<|>}`***<wks>***`|`***<portnum>***`]`** **`[action=pass|block]l`** Define a filter template with the specified parameters. Ip addresses and masks may be entered in either hexadecimal or in dotted-quad notation. 'tsport' and 'tdport' are transport-protocol source and destination ports, and can only be used if 'ipproto' matches against TCP or UDP. The action parameter defaults to "block".
`ip-fil-template|ift del|d` Delete the definition for template <*num*>.	**`template undefine`** *<template-number>* Delete the definition for template <*num*>.
...no available help	**`traceroute [-m max_ttl [-P UDP|TCP] [-p port] [-q nqueries] [-s src_addr] [-w waittime] host [datasize]`** Prints the route packets take to network host.

Table A.9 - IP / RIP Subsystem Commands

Old User Interface Commands	New User Interface Commands
...no available help	**`nenable|ndisable ad`** *<ifaddr>* Controls acceptance of default routes in updates sent to this net.
...no available help	**`nenable|ndisable auth`** *<ifaddr>* Generates and requires authentication in rip2 updates sent to this net.

Table A.9 - IP / RIP Subsystem Commands

Old User Interface Commands	New User Interface Commands
...no available help	**`nset auth [-k`** *<keyid>*|*<password>***`]`** *<ifaddr>* Set authorization string or key identifier on the specified vlan.
...no available help	**`nunset auth`** *<ifaddr>* Clears the authorization string on the specified vlan.
...no available help	**`nset authtype simple|s | md5|m`** *<ifaddr>* Set authorization type on the specified vlan.
`showcfg | scf` Show the configuration of the IP RIP subsystem.	**`config show`** Show the configuration of the IP RIP subsystem.
`ip-fil-acs-ctrl|ifa show|s` Display the ordered list of templates for both import and export filters. Templates which have been deleted are shown in (parenthesis). These deleted templates are ignored during lookup.	**`filter [show]`** Display the ordered list of templates for both import and export filters. Templates which have been deleted are shown in (parenthesis). These deleted templates are ignored during lookup.
`ip-fil-rule|ifr add|a` Define the given filter to be the ordered list of specified templates.	**`filter define import|export`** *<template-number>***`[,`***<template-number>***`...]`** Define the given filter to be the ordered list of specified templates.
`ip-fil-rule|ifr del|d` Delete the definition for the given filter.	**`filter undefine import|export`** Delete the definition for the given filter.
...no available help	**`filter append import|export`** *<template-number>***`[,`***<template-number>***`...]`** Adds the list of templates to the end of the given filter, which must already be defined.

Table A.9 - IP / RIP Subsystem Commands

Old User Interface Commands	New User Interface Commands
...no available help	**`filter insert import|export before|after`** *<template-number>*`|all` *<template-number>*`[,`*<template-number>*`...]` Inserts the list of templates into the given filter before or after the specified template. 'before all' prepends the new templates to the current list, while 'after all' appends them.
...no available help	**`filter delete import|export`** *<template-number>*`[,`*<template-number>*`...]` Removes the given templates from the specified filter.
...no available help	**`nenable|ndisable listen`** *<ifaddr>* Turn on or off acceptance of rip updates from this net.
...no available help	**`nenable|ndisable poison`** *<ifaddr>* Turn on or off "poison reverse" for updates sent to this net.
...no available help	**`nenable|ndisable rd`** *<ifaddr>* Controls reporting of default routes in updates sent to this net.
...no available help	**`nenable|ndisable rs`** *<ifaddr>* Controls reporting of static routes in updates sent to this net.
...no available help	**`nset rxtype rip1|rip2|both`** *<ifaddr>* Sets receive type on the specified vlan. **`rip1`**: RIP-1 only **`rip2`**: RIP-2 only **`both`**: Both RIP-1 and RIP-2
stats | s Shows statistics for the IP RIP subsystem.	stats [show] [-t] Shows statistics for the IP RIP subsystem.
stats-clear | sc Clears statistics for the IP RIP subsystem.	stats clear Clears statistics for the IP RIP subsystem.

Table A.9 - IP / RIP Subsystem Commands

Old User Interface Commands	New User Interface Commands
`ip-fil-template|ift show|s` Displays how many times the template has been successfully matched, since the last clear, both for import and for export. If no <*template-number*> is specified, show counts for all templates.	**stats [show] template [**<*template-number*>**]** Displays how many times the template has been successfully matched, since the last clear, both for import and for export. If no <*template-number*> is specified, show counts for all templates.
`ip-fil-template|ift del|d` Clears template statistics. Resets all template-match counts to zero.	stats clear template Clears template statistics. Resets all template-match counts to zero.
...no available help	**template [show] stats [**<*template-number*>**]** Displays how many times the template has been successfully matched, since the last clear, both for import and for export. If no <*template-number*> is specified, show counts for all templates.
...no available help	template clear stats Clears template statistics. Resets all template-match counts to zero.
...no available help	**nenable|ndisable talk** <*ifaddr*> Turns on or off generation of RIP updates for this net.
...no available help	**nset txtype rip1|rip1c|rip2** <*ifaddr*> Sets transmit type on the specified vlan. **`rip1`**: RIP-1 messages are sent **`rip1c`**: RIP-2 messages are broadcast **`rip2`**: RIP-2 messages are multicast
`ip-fil-template|ift show|s` Displays template contents. If no <*template-number*> is specified, show template contents for all templates.	**template [show] [**<*template-number*>**]** Displays template contents. If no <*template-number*> is specified, show template contents for all templates.

Table A.9 - IP / RIP Subsystem Commands

Old User Interface Commands	New User Interface Commands
ip-fil-template|ift chng|c Define a filter template with the specified parameters. Ip addresses and masks may be entered in either hexadecimal or in dotted-quad notation. 'tsport' and 'tdport' are transport-protocol source and destination ports, and can only be used if 'ipproto' matches against TCP or UDP. The action parameter defaults to "block".	**template define** *<template-number>* **[rif=***<ipaddr>***] [target=***<ipaddr>***/***<mask>***] [gw=***<ipaddr>***/***<mask>***] [tif=***<ipaddr>***] [sproto=static|interface|rip|ospf] [tag=***<tag>***]|[tag!=***<tag>***][tseg=***<seglist>***] action=[pass|block][,tag:***<tag>***][,metric:***<metric>***]** Defines a route-filter template with the specified parameters.
ip-fil-template|ift del|d Delete the definition for template <*num*>.	**template undefine** *<template-number>* Deletes the definition for template <*num*>.
...no available help	**template [show] stats [***<template-number>***]** Displays template contents. If no <*template-number*> is specified, show template contents for all templates.
...no available help	template clear stats Clears template statistics. Resets all template-match counts to zero.

Table A.10 - IP / OSPF Subsystem Commands

Old User Inerface Command	New User Interface Commands
area | ar Adds a new area to the OSPF Router.	**area add** *<area-id>* **[***<auth-type>***] [stub-area|sa** *<cost>***]** Adds a new area to the OSPF Router. Entering the optional stub-area-cost|sac will cause <*area-id*> to be added as a stub area with cost <*cost*>. **Note: The backbone area "0.0.0.0" is always present.
area | ar Deletes an area to the OSPF Router.	**area delete|del** *<area-id>***|all** Deletes the requested area or all areas. **Note: The backbone area, "0.0.0.0" can not be deleted.

Table A.10 - IP / OSPF Subsystem Commands

Old User Inerface Command	New User Interface Commands
`area | ar` Displays OSPF area information.	**`area [show|sh] [`*<area-id>*`]`** Displays OSPF area information. **`Area ID —`** A 32 bit integer uniquely identifying this area. Area "0.0.0.0" is the OSPF backbone. **`Auth`** - Authentication Type for this area. no = no authentication, sp = simple password. **`Imp AS Ext`** - Whether this area imports autonomous system external link-state advertisements, either Enabled or Disabled. **`SPF Runs`** - The number of times this area's intra-area route table has been calculated. **`Area Bdr`** - The number of reachable Area Border Routers. **`AS Bdr`** - The number of reachable Autonomous System Border Routers. **`Area LSAs`** - The total number of Link-State Advertisements in this area's LSA database excluding external LSAs. **`Stub Cost`** - Metric if this is a stub area, otherwise "-----".
...no available help	**`asbd enable|disable`** Enables or disables this router as an Autonomous System Border Router.
...no available help	**`asbd [show]`** Enables or disables this router as an Autonomous System Border Router.
`virtual-link | vl` Enables or disables the automatic virtual link feature.	**`auto-vlink enable|disable`** Enables or disables the automatic virtual link feature.
`auto-vlink [show]` Enables or disables the automatic virtual link feature.	**`auto-vlink [show]`** Enables or disables the automatic virtual link feature.

Old/New User Interface Commands

Table A.10 - IP / OSPF Subsystem Commands

Old User Inerface Command	New User Interface Commands
`showcfg | scf` Displays current OSPF Router configuration.	**`config [show]`** Displays current OSPF Router configuration.

`interface | if`
`Displays the interface table.`
	`interface|it [show] [`*<ip-addr>***`]`**
	`IP Address` - The IP address of this OSPF interface.
	`Area ID` - A 32-bit integer uniquely identifying the area to which this interface connects. Area ID "0.0.0.0" is the OSPF backbone.
	`DR` - The IP Address of the Designated Router.
	`BDR` - The IP Address of the Backup Designated Router.
	`Admin` - The Administrative Status of this interface.

...no available help	**`filter [show]`** Displays the ordered list of templates for the OSPF export filter. Templates which have been deleted are shown in (parenthesis). These deleted templates are ignored during lookup.
...no available help	**`filter define export`** *<template-number>***`[,`***<template-number>***`...]`** Defines the given filter to be the ordered list of specified templates. When exporting a route, each template is tested in increasing numerical order. Any template that matches will have its actions executed. The first action that includes 'pass' or 'block' will terminate the search. For OSPF, only "external" routes can be filtered, so every template must match upon 'sproto=rip'. They can also match upon *<target>*, *<gw>*, and *<tag>*. Other match criteria is invalid.
...no available help	**`filter undefine export`** Deletes the definition for the given filter.

Table A.10 - IP / OSPF Subsystem Commands

Old User Inerface Command	New User Interface Commands
...no available help	**`filter append export`** *<template-number>*`[,`*<template-number>*`...]` Adds the list of templates to the end of the given filter, which must already be defined.
...no available help	**`filter insert export before|after`** *<template-number>*`|all` *<template-number>*`[,`*<template-number>*`...]` Inserts the list of templates into the given filter before or after the specified template. 'before all' prepends the new templates to the current list, while 'after all' appends them.
...no available help	**`filter delete export`** *<template-number>*`[,`*<template-number>*`...]` Removes the given templates from the specified filter.
`show|sh` Displays Link State Advertisements. The optional parameters will display detailed information about an advertisement.	**`lsdb [show]`** *[<lsdbid> <rid> <type> <aid>]* Displays Link State Advertisements. The optional parameters will display detailed information about an advertisement. **`Area ID`** **-** The 32 bit Area ID from which the LSA was received. **`Lsdb Type`** **-** Type of Link State Advertisement. Options are: routerLink, networkLink, summaryLink, asSummaryLink, asExternalLink. **`Link State ID`** **-** Link State ID is either a Router ID or an IP address. **`Router ID`** **-** The ID of the router from which this LSA was received. **`Sequence`** **-** The OSPF sequence number is a 32 bit signed integer.

Table A.10 - IP / OSPF Subsystem Commands

Old User Inerface Command	New User Interface Commands
`show|sh` Displays non-virtual neighbors information in column format.	**`neighbor [show]`** Displays non-virtual neighbors information in column format. **`IP Address`** - The IP Address of this neighbor. **`Router ID`** - The OSPF Router ID of this neighbor. **`Pri`** - Priority used during Designated Router election. A value of 0 denotes that this router is ineligible to become the Designated Router. **`State`** - State of relationship with this router. States are: down, attempt, init, two Way, ex start (Exchange Start), exchange, loading, full. **`Events`** - The number of times this neighbor relationship has changed state or an error occurred. **`RTrQ`** - Current length of the retransmission queue.
`net-range | nr` Adds a network range to the specified area.	**`net-range add`** *<area-id> <net> <mask>* **`[noadv|na]`** Adds a network range to the specified area.
`net-range | nr` Deletes a network range from the specified area.	**`net-range delete|del`** *<area-id> <net> <mask>* Deletes a network range from the specified area.
`net-range | nr` Displays address range summaries to be propagated from an area.	**`net-range delete|del`** *<area-id> <net> <mask>* Displays address range summaries to be propagated from an area. **`Area ID`** - A 32-bit integer uniquely identifying the area to which this net range belongs. Area ID "0.0.0.0" is the OSPF backbone. **`Net`** - The IP address of the Net or Subnet. **`Mask`** - The Subnet Mask. **`Advertisement`** - Enable or disable advertising this net-range.

Table A.10 - IP / OSPF Subsystem Commands

Old User Inerface Command	New User Interface Commands
`set|se` Sets the OSPF router ID to <*router-id*>.	**`router-id set`** <*router-id*> Sets the OSPF router ID to <*router-id*>.
`set|se` Displays OSPF router-ids.	**`router-id [show]`** Displays OSPF router-ids.
`stats | s` Display statistics for OSPF.	**`stats [show]`** Display statistics for OSPF.
`stats | s` Clears statistics for OSPF.	**`stats clear`** Clears statistics for OSPF.
...no available help	**`stats [show] template`** Displays how many times the template has been successfully matched since the last clear. If no <*template-number*> is specified, show counts for all templates.
...no available help	**`stats clear template`** Clears template statistics. Resets all template-match counts to zero.
...no available help	**`template [show] [`**<*template-number*>**`]`** Displays template contents. If no <*template-number*> is specified, show template contents for all templates.

Table A.10 - IP / OSPF Subsystem Commands

Old User Inerface Command	New User Interface Commands
`ip-fil-template|ift chng|c` Define a filter template with the specified parameters. Ip addresses and masks may be entered in either hexadecimal or in dotted-quad notation. 'tsport' and 'tdport' are transport-protocol source and destination ports, and can only be used if 'ipproto' matches against TCP or UDP. The action parameter defaults to "block".	**template define** *<template-number>* **[rif=***<ipaddr>***] [target=***<ipaddr>***/***<mask>***] [gw=***<ipaddr>***/***<mask>***] [tif=***<ipaddr>***] [sproto=static|interface|rip|ospf] [tag=***<tag>***]|[tag!=***<tag>***] [tseg=***<seglist>***] action=[pass|block][,tag:***<tag>***][,metric:***<metric>***]** Defines a route-filter template with the specified parameters. IP addresses and masks are entered as 192.168.0.0/255.255.0.0, 10.1.2.3/h (host route), or 10.0.0.0/8 (equal to 10.0.0.0/255.0.0.0). *<tag>* is a 32-bit hex number, optionally starting with '0x'. On export route-filters, *<metric>* can be used to modify the outgoing metric. It can be a constant, one or two arithmetic operations, or a colon-separated array used as a lookup table.
`ip-fil-template|ift del|d` Delete the definition for template <*num*>.	**template undefine** *<template-number>* Deletes the definition for template <*num*>.
...no available help	**template [show] stats** *[<template-number>]* Display how many times the template has been successfully matched since the last clear. If no *<template-number>* is specified, show counts for all templates.
...no available help	**template clear stats** Clear template statistics. Resets all template-match counts to zero.

Table A.10 - IP / OSPF Subsystem Commands

Old User Inerface Command	New User Interface Commands
virtual-link | vl Manually add a virtual link.	**virtual-link|vlink add** *<aid> <router-id>* **[auth|a** *<key-str>***] [xdelay|x** *<trans-dly>***] [rint|r** *<rxmt-int>***] [hint|h** *<hello-int>***] [rdint|d** *<rtr-dead>***]** Manually adds a virtual link. Note: The transit area must be configured before adding the link.
virtual-link | vl Delete the requested virtual link.	**virtual-link|vlink delete|de***l* *<aid> <router-id>* Deletes the requested virtual link.
virtual-link | vl Displays Virtual Link information. Optional parameters will display detailed information about the link for that neighbor.	**virtual-link|vlink [show] [***<aid> <router-id>***]** Displays Virtual Link information. Optional parameters will display detailed information about the link for that neighbor. **Area ID -** A 32-bit integer uniquely identifying the area to which this virtual link belongs. Area ID "0.0.0.0" is the OSPF backbone. **Router ID -** The OSPF Router ID of this neighbor. **IP Address -** The IP Address of this neighbor. **If State -** The Virtual Interface State. Either up or down.

Table A.11 - IP Multicast Subsystem Commands

Old User Inerface Command	New User Interface Command
display-routecache | dc Shows the IP multicast routing cache.	cache [show] Shows the IP multicast routing cache.
flush-routecache | fc Flushes the IP multicast routing cache.	cache clear Flushes the IP multicast routing cache.

Table A.11 - IP Multicast Subsystem Commands

Old User Inerface Command	New User Interface Command
showcfg | scf Displays IP multicast routing configuration constants.	**config [show]** Displays IP multicast routing configuration constants.
getmem Allocates memory for the IP multicast routing table.	**getmem [2k|8k]** Allocates memory for the IP multicast routing table. By default, memory for 4k routes is allocated. The optional parameter allows specification of 2k or 8k routes.
add-interface | ai Adds a virtual interface, which maps to a configured physical interface.	**it|interface add** *<ipaddr>* **[met[ric]** *<metric>***] [thresh[old]** *<thresh>***]** Adds a virtual interface, which maps to a configured physical interface.
del-interface | di Deletes the virtual interface that maps to a configured physical interface.	**it|interface delete** *<ipaddr>***|all** Deletes the virtual interface that maps to a configured physical interface.
interface-table | it Shows the list of configured virtual interfaces.	**it|interface [show]** *[<disprestrictors>***]** Shows the list of configured virtual interfaces (both physical interfaces and tunnels).
show-mcast-groups | smg Shows the requested multicast address groups known to this router.	**multicast-groups|mg [show]** Shows the requested multicast address groups known to this router.
set|se Enables or disables the selective forwarding of multicast packets within a virtual LAN.	**enable|disable multicast-aware-bridging|mab** Enables or disables the selective forwarding of multicast packets within a virtual LAN. If this feature is disabled, multicast packets are forwarded to all segments in the VLAN. When this feature is enabled, multicast packets are forwarded based on reception of membership reports.
show-neighbors | sn Shows all the neighboring routers currently known to this router.	**neighbors [show]** Shows all the neighboring routers currently known to this router.

Table A.11 - IP Multicast Subsystem Commands

Old User Inerface Command	New User Interface Command
set|se Enables or disable the IP multicast pruning feature.	**enable|disable pruning** Enables or disable the IP multicast pruning feature.
route-table|rt Displays the requested IP multicast routing table (in column format).	**route|rt [show] [-c|-r] [-d] [-t] [-s] [***<disprestrictors>***]** Displays the requested IP multicast routing table (in column format).
flush-routecache | fc Flush all the dynamically learned IP multicast routing table entries.	**route|rt clear** Flush all the dynamically learned IP multicast routing table entries.
stats | s Displays the packet and error statistics for IGMP or Routing or DVMRP Default is the count since last statistics clear.	**stats [show] [-t] [dvm|igmp|rt|all]** Displays the packet and error statistics for IGMP or Routing or DVMRP Default is the count since last statistics clear.
stats-clear | sc Clears the packet and error statistics for IGMP or Routing or DVMRP.	**stats clear dvm|igmp|rt|all** Clears the packet and error statistics for IGMP or Routing or DVMRP.
...no available help	**penable|pdisable transmit** *<segment-list>* Enables or disables sending multicast datagrams over ports in *<segment-list>*.
add-tunnel | at Add a virtual interface, which maps to a tunnel between this router and the router whose address is *<remote-addr>*.	**tunnel add [-s] loc[al]** *<locadr>* **rem[ote]** *<rmtadr>* **[met[ric]** *<mv>***] [thresh[old]** *<tv>***]** Add a virtual interface, which maps to a tunnel between this router and the router whose address is *<remote-addr>*.
del-tunnel | dt Deletes the a virtual interface which maps to a tunnel between this router and the router with the address *<remote-addr>*.	**tunnel delete (loc[al]** *<local-addr>* **rem[ote]** *<remote-addr>***) | all** Deletes the a virtual interface which maps to a tunnel between this router and the router with the address *<remote-addr>*. "All" deletes all tunnels.

Table A.12 - IPX Subsystem Commands

Old User Inerface Command	New User Interface Commands
display-routecache | dc Displays the route cache.	**cache [show] [<*disprestrictions*>]** Displays the route cache.
showcfg | scf Displays generic IPX configuration information.	config [show] Displays generic IPX configuration information.
...no available help	[set] diag value Sets internal IPX diagnostic mask to value.
getmem Allocates memory for IPX routing.	getmem Allocates memory for IPX routing.
enable | disable helper Enables or disables the IPX helper feature.	enable | disable helper Enables or disables the IPX helper feature.
helper | helper Adds a helper address to the specified segment(s).	**helper add** <*seglist*> <*network*> <*node address*> **s[ock[et]]** <*socket*> Adds a helper address to the specified segment(s). Use FFFFFFFF for <*network*> to specify all nets. <*node address*> can be a unicast address, or a broadcast address.
helper | helper Deletes the helper address set up on the specified segment(s).	**helper del[ete]** <*seglist*> Deletes the helper address set up on the specified segment(s).
helper | helper Shows the currently set up helper address for all segments.	helper [show] Shows the currently set up helper address for all segments.
add-interface | ai Adds an IPX interface to the given segment(s).	**it | interface add** <*segmentlist*> <*network*> **[mtu** <*mtu*>**] [met[ric]** <*metric*>**] [encap enet | 802.3 | 802.2 | snap]** Adds an IPX interface to the given segment(s).
del-interface | di Deletes (possibly several) IPX interfaces.	**it | interface del[ete]** <*seglist*> <*network*> **| all** Deletes (possibly several) IPX interfaces.
interface-table | it Shows the list of configured IPX interfaces.	**it | interface [show] [<*disprestrictors*>]** Shows the list of configured IPX interfaces.

Table A.12 - IPX Subsystem Commands

Old User Inerface Command	New User Interface Commands
rip-pset | rpse This command will select the mode for sending and receiving RIP updates	set ripsap-ctrl | rsct normal | n | vlan | v This command will select the mode for sending and receiving RIP updates. Selecting "normal" will cause the system to use per-port control parameters (which are set using the "set" command.) Selecting "vlan" will cause the system to use per network control parameters (which are set using the "rip-ctrl-tbl" command.) If no argument is entered, the current control type is displayed.
ripsap-ctrl-type | rsct Displays the selected mode for sending and receiving RIP updates.	[show] ripsap-ctrl | rsct Displays the selected mode for sending and receiving RIP updates.
route-table | rt Displays routing table.	**route | rt [show] [-c | -r | -t] [**<*disprestrictors*>**]** Displays routing table.
**`server-table	st`** Displays the server table.	**server [show] [-f | -a | -t] [**<*disprestrictors*>**]** <*disprestrictors*> **=** **s[eg[ment]]=**<*seglist*> **n[et[work]]=**<*network*> **na[me]=**<*name*> **ty[pe]=**<*type*> Displays the server table.
stats | s Displays packet stats for IPX packets.	stats [show] [-t] Displays packet stats for IPX packets.
stats-clear | sc Clear IPX packet statistics.	stats clear Clear IPX packet statistics.
set | se Enables or disables the IPX type 20 packet forwarding.	enable | disable type20-forwarding | t20fw Enables or disables the IPX type 20 packet forwarding.
stats | s Displays packet stats for Type20 packets.	t20stats [show] [-t] Displays packet stats for Type20 packets.
stats-clear | sc Clears Type20 packet statistics.	t20stats clear Clears Type20 packet statistics.

Table A.12 - IPX Subsystem Commands

Old User Inerface Command	New User Interface Commands
set | se Enables or disables Type 20 packet forwarding for the specified segments.	**penable|pdisable type20-port-forwarding|tpfw** *<seglist>* Enables or disables Type 20 packet forwarding for the specified segments.
stats|s Displays packet stats for Type20 packets.	**type20-port-forwarding|tpfw [show]** *<seglist>* Shows the current setup of the specified segments.

Table A.13 - IPX/RIP Subsystem Commands

Old User Inerface Command	New User Interface Commands
showcfg | scf Displays RIP config info for segments.	**config [show] [***<seglist>***|all]** Displays RIP config info for segments.
dif|data-input-filter [show] [*<fnum-list>***|all]** Displays configured rip data input filters.	**dif|data-input-filter [show] [***<fnum-list>***|all]** Displays configured rip data input filters.
ospf-import-filter | oif Adds a new rip data input filter.	**dif|data-input-filter add** *<fnum>* **block|pass** *<targetnet> <rxnet>* Adds a new rip data input filter.
ospf-import-filter | oif Deletes specified filters.	**dif|data-input-filter delete** *<fnum-list>***|all** Deletes specified filters.
ospf-export-filter|oef Displays configured rip data output filters.	**dof|data-output-filter [show] [***<fnum-list>***|all]** Displays configured rip data output filters.
ospf-export-filter|oef Adds a new rip data output filter.	**dof|data-output-filter add** *<fnum>* **block|pass** *<targetnet> <txnet>* Adds a new rip data output filter.

Table A.13 - IPX/RIP Subsystem Commands

Old User Inerface Command	New User Interface Commands
ospf-export-filter | oef Deletes specified filters.	**dof | data-output-filter delete** *<fnum-list>* **| all** Deletes specified filters.
rip-update-filter | ruf Enables or disables acceptance of rip packets on segments in *<seglist>*.	**penable | pdisable li[sten]** *<seglist>* **| all** Enables or disables acceptance of rip packets on segments in *<seglist>*.
rip-update-filter | ruf Enables or disables acceptance of rip packets on *<network>*.	**nenable | ndisable li[sten]** *<network>* Enables or disables acceptance of rip packets on *<network>*.
...no available help	**pof | pkt-output-filter [show] [***<fnum-list>* **| all]** Displays configured rip packet output filters.
...no available help	**pof | pkt-output-filter add** *<fnum>* **block | pass** *<segment> <txnet>* Adds a new rip packet output filter.
...no available help	**pof | pkt-output-filter delete** *<fnum-list>* **| all** Deletes specified filters.
stats | s Displays RIP packet statistics.	stats [show] [-t] Displays RIP packet statistics.
stats-clear | sc Clears RIP packet statistics.	stats clear Clears RIP packet statistics.
penable | pdisable ta[lk] *<seglist>* **| all** Enables or disables generation of rip packets on segments in *<seglist>*.	**penable | pdisable ta[lk]** *<seglist>* **| all** Enables or disables generation of rip packets on segments in *<seglist>*.
...no available help	**nenable | ndisable ta[lk]** *<network>* Enables or disables generation of rip packets on *<network>*.
...no available help	**set timers** *<transmit-intvl>* **[***<rip-age>***]** Sets RIP transmit timer and age timer values.

Table A.14 - IPX/SAP Subsystem Commands

Old User Inerface Command	New User Interface Commands
sap-showcfg | sscf Displays generic IPX configuration information.	**config [show] [<*disprestrict*>]** Displays generic IPX configuration information.
sap-report-filter | srf Displays configured sap data input filters.	**sap dif | data-input-filter show [-f] [<*fnum-list*> | all]** Displays configured sap data input filters.
sap-report-filter | srf Adds a new sap data input filter.	**sap dif | data-input-filter add** <*fnum*> **block | pass** <*stype*> <*sname*> <*rxnet*> Adds a new sap data input filter.
sap-report-filter | srf Deletes specified filters.	**sap dif | data-input-filter delete** <*fnum-list*> **| all** Deletes specified filters.
...no available help	**sap dof | data-output-filter show [-f] [**<*fnum-list*> **| all]** Displays configured sap data output filters.
...no available help	**sap dof | data-output-filter add** <*fnum*> **block | b | block-nearest | bn | pass | p** <*stype*> <*sname*> <*txnet*> Adds a new sap data output filter. "block-nearest" action will hide a server only in the response to a get-nearest-server request. The server is reported to other routers in regular SAP responses.
...no available help	**sap dof | data-output-filter delete** <*fnum-list*> **| all** Deletes specified filters.
sap-pset | spse Enables or disables acceptance of sap packets on segments in <seglist>.	**penable | pdisable li[sten]** <*seglist*> **| all** Enables or disables acceptance of sap packets on segments in <seglist>.
sap-pset | spse Enables or disables acceptance of sap packets on <*network*>.	**nenable | ndisable li[sten]** <*network*> Enables or disables acceptance of sap packets on <*network*>.

Table A.14 - IPX/SAP Subsystem Commands

Old User Inerface Command	New User Interface Commands
...no available help	**sap pof|pkt-output-filter show [**<*fnum-list*>**|all]** Displays configured sap packet output filters.
...no available help	sap pof|pkt-output-filter add <fnum> block|pass <segment> <txnet> Adds a new sap packet output filter.
...no available help	sap pof|pkt-output-filter delete <fnum-list>|all Deletes specified filters.
sap-stats | sst Displays packet stats for IPX packets.	stats [show] [-t] Displays packet stats for IPX packets.
sap-stats-clear | sstc Clears IPX packet statistics.	stats clear Clears IPX packet statistics.
sap-pset | spse Enables or disables generation of sap packets on segments in <*seglist*>.	**penable|pdisable ta[lk]** <*seglist*>**|all** Enables or disables generation of sap packets on segments in <*seglist*>.
sap-pset | spse Enables or disables generation of sap packets on <*network*>.	**nenable|ndisable ta[lk]** <*network*> Enables or disables generation of sap packets on <*network*>.
...no available help	**set timers** <*transmit-interval-time*> **[**<*aging-time*>**]** Sets SAP transmit timer and age timer values.

Table A.15 - SNMP Subsystem Commands

Old User Inerface Command	New User Interface Commands
add | a Add a management community from which requests may be received (up to 8 may be added). Default access is ro (read-only).	**community|com add** <*community-name*> **[ro|rw]** Add a management community from which requests may be received (up to 8 may be added). Default access is ro (read-only).

Table A.15 - SNMP Subsystem Commands

Old User Inerface Command	New User Interface Commands
delete | d Deletes a community (and the list of associated managers).	**community|com delete|del** *<community-name>* Deletes a community (and the list of associated managers).
showcfg|scf Display name and access level for added communities. If <community-name> is not given, display all communities. "-l" displays list of managers.	**config [show] [-l] [***<community-name>***]** Display name and access level for added communities. If <community-name> is not given, display all communities. "-l" displays list of managers.
add | a Add up to 16 managers per community. Traps enabled on per-manager basis. Default is trap, (trap sent to each manager added).	**manager|man add** *<community-name> <IP-addr>* **[trap|notrap]** Add up to 16 managers per community. Traps enabled on per-manager basis. Default is trap, (trap sent to each manager added).
delete | d Deletes a manager from a given community. If "all" is entered, all managers for a given community will be deleted.	**manager|man delete|del** *<community-name> <IP-addr>***|all** Deletes a manager from a given community. If "all" is entered, all managers for a given community will be deleted.
stats | s Displays SNMP packet statistics. Default is count since last statistics clear.	**stats [show] [-t]** Displays SNMP packet statistics. Default is count since last statistics clear.
stats-clear | sc Clears the SNMP packet statistics.	**stats clear** Clears the SNMP packet statistics.

Table A.16 - ATALK Subsystem Commands

Old User Inerface Command	New User Interface Commands
arp-table | at Shows the AARP table.	**arp [show] [***<disprestrict>***]** Shows the AARP table.
arp-tableclear | atc Clears dynamic entries from the AARP table.	arp clear Clears dynamic entries from the AARP table.

Table A.16 - ATALK Subsystem Commands

Old User Inerface Command	New User Interface Commands
set-arpage | saa Sets the AARP aging timer interval.	**arp set age** *<time>* Sets the AARP aging timer interval. <time> must be an integral number of minutes (i.e., a multiple of sixty seconds).
...no available help	arp [show] age Shows the AARP aging timer interval.
set atr enl | dis Enables or disables Appletalk routing.	enable | disable atalk Enables or disables Appletalk routing.
display-routecache | dc Displays the route cache.	**cache [show] [***<disprestrict>***]** Displays the route cache.
flush-routecache | fc Flushes the route cache.	cache clear Flushes the route cache.
showcfg | scf Displays current Appletalk configuration.	config [show] Displays current Appletalk configuration.
add-interface|a Adds a non-seed segment.	**interface | it add [-n]** *<seglist>* **-n** - non-appletalk / backbone / passive **-h** - hard-seed This form of the command adds a non-seed segment. The **-n** option may be specified to indicate a VLAN backbone segment.
add-interface|a Adds a seeding segment.	**interface | it add** *<seglist> <net>***.***<node>* **net[range]** *<x>***-***<y>* **-n** - non-appletalk / backbone / passive **-h** - hard-seed This form of the command adds a seeding segment with the specified net and node numbers.

Table A.16 - ATALK Subsystem Commands

Old User Inerface Command	New User Interface Commands
add-interface|a Adds a hard seeding segment with the specified net and node numbers.	**interface|it add [-h]** *<seglist> <net>***.***<node>* **net[range]** *<x>***-***<y>* **-n** - non-appletalk / backbone / passive **-h** - hard-seed This form of the commands adds a hard seeding segment with the specified net and node numbers. Zones (including the default zone) will be configured according to the zone configuration table (as set by the "zone add" command).
del-interface **Deletes one or more appletalk networks.**	**interface|it del[ete] [-a]** *<seglist>* **Deletes one or more appletalk networks.** **-a** - deletes all interfaces associated with the network numbers configured on the segments in *<seglist>*.
interface-table|it Shows network address range and zone names that are assigned to a set of segments. By default, the entire table is displayed.	**interface|it [show] [-c] [***<disprestrict>***]** Shows network address range and zone names that are assigned to a set of segments. By default, the entire table is displayed. **-a** - includes segments that haven't been set up with "it add" **-c** - shows configured interface information, not dynamic info **-z** - shows zones in abbreviated form
nbp-fwd-filter add Adds a forwarding filter for NBP lookups. These filters specify whether or not to forward NBP lookup requests for the given zone on the specified segment.	**nbp-fwd-filter|nff add** *<filnum>* **b[lock]|p[ass]** *<seg> <zone>* Adds a forwarding filter for NBP lookups. These filters specify whether or not to forward NBP lookup requests for the given zone on the specified segment.

Table A.16 - ATALK Subsystem Commands

Old User Inerface Command	New User Interface Commands
nbp-fwd-filter del | d Adds a forwarding filter for NBP lookups. These filters specify whether or not to forward NBP lookup requests for the given zone on the specified segment.	**nbp-fwd-filter | nff delete** <*filnum*> **| all** Deletes an NBP forwarding filter.
nbp-fwd-filter | nff show | s Displays NBP forwarding filters.	**nbp-fwd-filter | nff [show] [**<*filnum*>**[,**<*filnum*>**. . .]]** Displays NBP forwarding filters.
name-table | nt Displays the table of NBP services registered on this entity.	name/nt [show] Displays the table of NBP services registered on this entity.
ping Ping a specific AppleTalk address.	**ping [-t** <*timeout*>**] [-size** <*size*>**]** <*net*>**.**<*node*> Ping a specific AppleTalk address. By default the system waits for 15 seconds before timing out unless a time out value is specified. The default packet size is 64 bytes. This can be changed (via -size) to any value between 64 and 586 bytes inclusive.
route-table | rt Displays the routing table.	**route | rt [show] [-c | -r] [**<*disprestrict*>**]** Displays the routing table. **-c** - directly-connected routes only **-r** - RTMP entries only **-t** - totals only
stats | s Displays statistics for AppleTalk ARP, DDP, or echo packets.	stats [show] [arp | ddp | echo] Displays statistics for AppleTalk ARP, DDP, or echo packets.
stats-clear | sc Clear statistics for AppleTalk ARP, DDP, or echo packets.	stats clear arp | ddp | echo Clear statistics for AppleTalk ARP, DDP, or echo packets.

Table A.16 - ATALK Subsystem Commands

Old User Inerface Command	New User Interface Commands
zone-accept-filter|zaf add|a Specifies zones to be ignored from inbound ZIP reports for this netrange.	**zone-data-input-filter|zdif add** *<filnum>* **b[lock]|p[ass]** *<netrange>***|all** *<zone>***|*** Specifies zones to be ignored from inbound ZIP reports for this netrange.
zone-accept-filter|zaf del|d Deletes a zone-data-input filter.	**zone-data-input-filter|zdif delete** *<filnum>* Deletes a zone-data-input filter.
zone-accept-filter|zaf show|s Displays zone-data-input filters.	**zone-data-input-filter|zdif [show] [***<filnum>***[,***<filnum>***...]]** Displays zone-data-input filters.
...no available help	**zone-data-output-filter|zdof add** *<filnum>* **b[lock]|p[ass]** *<netrange>***|all <zone>|*** Specifies zones to be dropped from outbound ZIP reports for this netrange.
...no available help	**zone-data-output-filter|zdof delete** *<filnum>***|all** Deletes a zone-data-output filter.
...no available help	**zone-data-output-filter|zdof [show] [***<filnum>***[,***<filnum>***...]]** Displays zone-data-output filters.
add-zone | az Adds a zone name to the zone configuration table.	**zone|zt add [-d]** *<net-range> <zone>* Adds a zone name to the zone configuration table. A zone name is a string of 32 characters that are case insensitive. Zone names may contain spaces; use quotes to enclose such zone names. The zone configuration table holds a list of zones for each netrange which are used when seeding that netrange. **-d** - The zone should be the default zone for this netrange.

Table A.16 - ATALK Subsystem Commands

Old User Inerface Command	New User Interface Commands
zone-table | zt Displays a table of active zone names.	zone|zt [show] [-c] [<*disprestrict*>] Displays a table of active zone names. A "*" before the zone name indicates the default zone name for this net-range. **-c** - show zone configuration table instead.
del-zone | dz Deletes one or more zone names from the zone configuration table.	zone|zt delete <*net-range*> <*zone*> Deletes one or more zone names from the zone configuration table. A zone name is a string of 32 characters that are case insensitive. Zone names may contain spaces; use quotes to enclose such zone names.
zone-update-filter|zuf add|a Adds a zone-pkt-output filter. These filters specify whether or not to send any zone data packets on the given <*seg*>/<*netrange*> combination.	**zone-pkt-output-filter|zpof add** <*filnum*> **b[lock]|p[ass]** <*seg*> <*netrange*> Adds a zone-pkt-output filter. These filters specify whether or not to send any zone data packets on the given <*seg*>/<*netrange*> combination.
zone-update-filter|zuf del|d Deletes a zone-pkt-output filter.	**zone-pkt-output-filter|zpof delete** <*filnum*>**|all** Deletes a zone-pkt-output filter.
zone-update-filter|zuf show|s Displays zone-pkt-output filters.	**zone-pkt-output-filter|zpof [show] [**<*filnum*>**[,**<*filnum*>**...]]** Displays zone-pkt-output filters.

Table A.17 - DECnet Subsystem Commands

Old User Inerface Command	New User Interface Commands
display-endnode-adj | dea This form displays adjacencies to all the end nodes, or to the selected end node.	**[show] adj[acent] [end]node[s] [[a[ddr]=]**<*node*>**]** This form displays adjacencies to all the end nodes, or to the selected end node. <*node*> must be between 1 and 1023. Additionally, <*node*> must not exceed the maximum node number (as specified by the max-node-num variable, default 255).
display-router-adj | dra This form shows adjacencies to all the routers, or the specified node.	[show] adj[acent] r[outer[s]] [[a[ddr]=]<node>] This form shows adjacencies to all the routers, or the specified node. <*node*> must be between 1 and 1023. Additionally, <*node*> must not exceed the maximum node number (as specified by the max-node-num variable, default 255).
display-area-tbl | dat Displays routes to all the reachable areas, or to specified area. <*area*> must be between 1 and 63.	**area [show] [**<*area*>**]** Displays routes to all the reachable areas, or to specified area. <*area*> must be between 1 and 63. This command is applicable only if the node is an area router.
set-port-param|spp Displays the data link block size for the port(s) in <*seglist*>.	**show block-size|bs [**<*seglist*>**]** Displays the data link block size for the port(s) in <*seglist*>.
set-port-param|spp Displays the data link block size for the port(s) in <*seglist*>.	**pset block-size|bs** <*value*> <*seglist*> Sets the data link block size for the port(s) in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all".
display-routecache | dc Display DECnet route cache. <*seglist*> is a comma-separated list of ports or "all".	**cache [show] [**<*disprestrict*>**]** Display DECnet route cache. <*seglist*> is a comma-separated list of ports or "all".
flush-routecache | fc Clears DECNET routing cache.	cache clear Clears DECNET routing cache.

Table A.17 - DECnet Subsystem Commands

Old User Inerface Command	New User Interface Commands
set-port-param|spp cost|cos Set the cost for the ports in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all". Range for <*value*>: 1-127	**pset cost|c** <*value*> <*seglist*> Set the cost for the ports in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all". Range for <*value*>: 1-127
...no available help	**show cost [**<*seglist*>**]** Shows port cost.
getmem Gets memory for Decnet routing.	getmem Gets memory for Decnet routing.
...no available help	**show hello-time|ht [**<*seglist*>**]** Displays the interval for sending hello packets on the port(s) in <seglist>.
set-port-param|spp hello-time|htm Set the interval for sending hello packets on the port(s) in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all". Range for <value>: 1-8191	**pset hello-time|ht** <*value*> <*seglist*> Set the interval for sending hello packets on the port(s) in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all". Range for <value>: 1-8191
display-endnode-adj | dea Sets the number of endnode adjacencies supported by this router.	**set max-adj-endnodes|mae** <*value*> Sets the number of endnode adjacencies supported by this router. Range for <value>: 1-1023.
display-endnode-adj | dea Displays the number of endnode adjacencies supported by this router.	show max-adj-endnodes|mae Displays the number of endnode adjacencies supported by this router.
display-router-adj | dra Set the number of broadcast router adjacencies supported by this router.	**set max-adj-routers|mar** <*value*> Set the number of broadcast router adjacencies supported by this router. Range for <value>: 1-560.
display-router-adj | dra Displays the number of broadcast router adjacencies supported by this router.	show max-adj-routers|mar Displays the number of broadcast router adjacencies supported by this router.

Table A.17 - DECnet Subsystem Commands

Old User Inerface Command	New User Interface Commands
`set-node-param|snp max-area-num | man` Sets the maximum area number allowed in the entire network. Range for <*value*>: 1 - 63. <*value*> must be greater than or equal to maximum area number in use.	**set max-area-num | man** <*value*> Sets the maximum area number allowed in the entire network. Range for <*value*>: 1 - 63. <*value*> must be greater than or equal to maximum area number in use.
`set-node-param|snp max-area-num | man` Displays the maximum area number allowed in the entire network.	show max-area-num | man Displays the maximum area number allowed in the entire network.
`set-node-param|snp max-cost-to-area | mca` Displays the maximum cost possible in a path to a reachable area.	show max-cost-to-area | mca Displays the maximum cost possible in a path to a reachable area.
`set-node-param|snp max-cost-to-area | mca` Sets the maximum cost possible in a path to a reachable area. Range for <*value*>: 1 - 1022. <*value*> must be greater than or equal to (actual max hops to an area * 25).	**set max-cost-to-area | mca** <*value*> Sets the maximum cost possible in a path to a reachable area. Range for <*value*>: 1 - 1022. <*value*> must be greater than or equal to (actual max hops to an area * 25).
`set-node-param|snp max-cost-to-node | mcn` Displays the maximum cost possible in a path to a reachable node.	show max-cost-to-node | mcn Displays the maximum cost possible in a path to a reachable node.
`set-node-param|snp` Displays the maximum cost possible in a path to a reachable node. Sets the maximum cost possible in a path to a reachable node. Range for <*value*>: 1 - 1022. <*value*> must be greater than or equal to (actual max hops in area * 25).	**set max-cost-to-node | mcn** <*value*> Sets the maximum cost possible in a path to a reachable node. Range for <*value*>: 1 - 1022. <*value*> must be greater than or equal to (actual max hops in area * 25).

Table A.17 - DECnet Subsystem Commands

Old User Inerface Command	New User Interface Commands
set-node-param|snp max-hops-to-area | mha Displays the maximum hops possible in a path to a reachable area.	show max-hops-to-area|mha Displays the maximum hops possible in a path to a reachable area.
set-node-param|snp max-hops-to-area | mha Sets the maximum hops possible in a path to a reachable area. Range for <*value*>: 1 - 30. <*value*> must be greater than or equal to actual max hops to any area.	**set max-hops-to-area|mha** <*value*> Sets the maximum hops possible in a path to a reachable area. Range for <*value*>: 1 - 30. <*value*> must be greater than or equal to actual max hops to any area.
set-node-param|snp max-cost-to-node | mcn Displays the maximum hops possible in a path to a reachable node.	show max-cost-to-node|mcn Displays the maximum hops possible in a path to a reachable node.
set-node-param|snp max-hops-to-node | mhn Sets the maximum hops possible in a path to a reachable node. Range for <*value*>: 1 - 30. <*value*> must be greater than or equal to actual max hops in an area.	**set max-hops-to-node|mhn** <*value*> Sets the maximum hops possible in a path to a reachable node. Range for <*value*>: 1 - 30. <*value*> must be greater than or equal to actual max hops in an area.
set-node-param|snp max-node-num | mnn Displays the maximum node number allowed within this area.	show max-node-num|mnn Displays the maximum node number allowed within this area.
set-node-param|snp max-node-num | mnn Sets the maximum node number allowed within this area. Range for <*value*>: 1 - 1023. <*value*> must be greater than or equal to maximum node number in use.	**set max-node-num|mnn** <*value*> Sets the maximum node number allowed within this area. Range for <*value*>: 1 - 1023. <*value*> must be greater than or equal to maximum node number in use.
...no available help	**show max-routers|mr [**<*seglist*>**]** Displays the number of broadcast router adjacencies supported on the port(s) in <*seglist*>.

Table A.17 - DECnet Subsystem Commands

Old User Inerface Command	New User Interface Commands
`set-port-param|spp max-routers|mrt` Sets the number of broadcast router adjacencies supported on the port(s) in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all". Range for <value>: 1-20.	**pset max-routers | mr** <*value*> <*seglist*> Sets the number of broadcast router adjacencies supported on the port(s) in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all". Range for <value>: 1-20.
`set-node-param|snp max-visits | mvs` Displays the maximum visits for a packet before the router assumes that the packet is looping.	show max-visits | mv Displays the maximum visits for a packet before the router assumes that the packet is looping.
`set-node-param|snp max-visits | mv` Sets the maximum visits for a packet before the router assumes that the packet is looping. Range for <*value*>: maxpath - 60. <*value*> must be greater than equal to the actual maximum path in the entire network.	**set max-visits | mv** <*value*> Sets the maximum visits for a packet before the router assumes that the packet is looping. Range for <*value*>: maxpath - 60. <*value*> must be greater than equal to the actual maximum path in the entire network.
`set-node-param|snp node-type | ntp` Displays the type of routing supported by this router.	show node-type | nt Displays the type of routing supported by this router.
`set-node-param|snp node-type | ntp` Sets the type of routing supported by this router. The options are: router | rt: level 1 routing area-router | ar: level 2 routing	**set node-type | nt** <*value*> Sets the type of routing supported by this router. The options are: router | rt: level 1 routing area-router | ar: level 2 routing
`set-node-param|sn node-id | nid` Displays the node identifier for this router.	show node-id | nid Displays the node identifier for this router.
`set-node-param|sn node-id | nid` Sets the node identifier for this router. <*area*> must not exceed the value of max-area-num <*node*> must not exceed the value of max-node-num. The lower bound for both values is 1.	**set node-id | nid** <*area*>**.**<*node*> Sets the node identifier for this router. <*area*> must not exceed the value of max-area-num <*node*> must not exceed the value of max-node-num. The lower bound for both values is 1.
`set-port-param|spp priority|pri` Displays the priority for the port(s) in <*seglist*>.	**show priority | pri [**<*seglist*>**]** Displays the priority for the port(s) in <*seglist*>.

Table A.17 - DECnet Subsystem Commands

Old User Inerface Command	New User Interface Commands
`set-port-param|spp priority|pri` Sets the priority for the port(s) in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all". Range for <value>: 0 - 127.	**pset priority | pri** <*value*> <*seglist*> Sets the priority for the port(s) in <*seglist*>. <*seglist*> is a comma-separated list of ports or "all". Range for <value>: 0 - 127.
display-route-tbl | drt Displays routes to all the reachable nodes, or to the specified node.	**route | rt [show] [**<*disprestrict*>**]** Displays routes to all the reachable nodes, or to the specified node. <*node*> must be between 1 and 1023. If "area-rtr" is specified, display route to nearest area router.
routing-status | rs Shows the status of DECnet-forwarding and the routing state of all of the segments.	routing-status | rs [show] Shows the status of DECnet-forwarding and the routing state of all of the segments.
port-stats-clear | psc Clears statistics.	**stats [clear]** <*params*> Clears statistics. <*params*> = p[ort] [<*seglist*>] | n[ode] <*seglist*> is a comma-separated list of ports or "all".
`port-stats | ps` Shows DECNET statistics.	**stats [show]** <*params*> Shows DECNET statistics. <*params*> = p[ort] [-t] <*seglist*> | n[ode] [-t] <*seglist*> is a comma-separated list of ports or "all".
...no available help	show update-time | ut Displays background timer for sending routing updates.
...no available help	**set update-time | ut** <*secs*> Sets background timer for sending routing updates. Range for <secs>: 1-1200.

Table A.18 - FDDI Subsystem Commands

Old User Inerface Command	New User Interface Commands
showConcentrator | shc Displays the current FDDI concentrator configuration.	**[show] concentrator|con [**<*slot-list*> **to** <*fddi-segment*>**]** Displays the current FDDI concentrator configuration.
concentratorAttach | cona Attaches one or more concentrator modules to the FDDI Dual attach concentrator.	**[attach] concentrator|con [**<*slot-list*> **to** <*fddi-segment*>**]** Attaches one or more concentrator modules to the FDDI Dual attach concentrator. Or add additional modules to the concentrator. <*concentrator-slot-list*> (using space as a deliminator) is a list of concentrator modules to be attached to the FDDI segment. The concentrator module must be in one of the following slots: * slot 3 to 7 in a 10-slot chassis <*fddi-segment*> is to be used as the Dual Attach Concentrator (DAC). The FDDI segment must be in one of the following slots: * slot 3 to 7 in a 10-slot chassis. Only one FDDI modules can be configured as a single DAC or dual DACs.
concentratorDetach | cond Detaches FDDI concentrator modules from <*seg-list*> or detach <*all*> of the FDDI concentrator modules.	**[detach] concentrator|con [**<*slot-list*> **to** <*fddi-segment*>**]** Detaches FDDI concentrator modules from <*seg-list*> or detach <*all*> of the FDDI concentrator modules.
showDAC | sdac Displays information about the FDDI Dual Attach Concentrator.	[show] dac Displays information about the FDDI Dual Attach Concentrator.
shownvram | shn Displays the FDDI Dual Attach Concentrator NVRAM.	[show] nvram Displays the FDDI Dual Attach Concentrator NVRAM.

Table A.18 - FDDI Subsystem Commands

Old User Inerface Command	New User Interface Commands
setmacparam|smp treq|tvx Sets the T_REQ parameter on the specified fddi port(s) to the indicated time value. The range for the time value is 4 - 167 milliseconds. Therefore, the time value must be specified with a trailing m (for milliseconds) or u (for microseconds).	**pset treq** *<time>* **| default** *<portlist>* Sets the T_REQ parameter on the specified fddi port(s) to the indicated time value. The range for the time value is 4 - 167 milliseconds. Therefore, the time value must be specified with a trailing m (for milliseconds) or u (for microseconds).
setmacparam|smp treq|tvx Sets the TVX parameter on the specified fddi card to the indicated time value. The range for the time value is 2621 - 5200 microseconds. Therefore, the time value must be specified with a trailing u (for microseconds).	**pset tvx** *<time>* **| default** *<portlist>* Sets the TVX parameter on the specified fddi card to the indicated time value. The range for the time value is 2621 - 5200 microseconds. Therefore, the time value must be specified with a trailing u (for microseconds).
showResetCt | src Displays the fddi reset count for all fddi port(s).	[show] resetct | src Displays the fddi reset count for all fddi port(s).
showsmtmib | ssm Displays the fddi smt-mib variables for the specified fddi port(s).	**[show] smtmib [***<group>***] [***<disprestrict>***]** Displays the fddi smt-mib variables for the specified fddi port(s). *<group>* is one of the following: **smt | mac | port | all**
showcounter | sct Counter status: Shows FDDI private counters of a FDDI board in the specified slot.	**status [show] counter** *<slot>* Counter status: Shows FDDI private counters of a FDDI board in the specified slot.

APPENDIX B Configuration Defaults

This appendix lists the PowerHub configuration defaults. The purpose of this appendix is to help understand what is already configured in the software to help in diagnosing and troubleshooting PowerHub problems. Configuration defaults are listed by subsytem.

Table B.1 - Boot PROM Commands

Command and Description
zreceive | zr | rz [-+27abcehtw] [<*file-name*>**]** t Sets the receive timeout to N / 10 seconds (10 <= N <= 1000). The default is **`100`**; that is, 10 seconds.
zsend | zs | sz [-+27abehkLlNnoptwXYy] <*file-name*> t Sets the receive timeout to N / 10 seconds (10 <= N <= 1000). The default is **600**; that is, 60 seconds.

Table B.2 - Global Subsystem Commands

Command and Description
su [root | monitor] Lets you change the userid to the root or monitor. The default is the root.
rm [-f] [-i] <*filespec*> **[**<*filespec*>...**]** Overrides the **`-f`** (Force) flag, presenting a prompt before removing each file. If you do not specify **`-f`** or **`-i`**, **`-i`** is the default.

Table B.3 - ATALK Subsystem Commands

Command and Description
enable | disable atalk Specifies whether you are enabling or disabling AppleTalk routing. The default is **disable**.
ping [-t <*timeout*>] [-size <*pktsize*>] <*net*>.<*node*> [-t <*timeout*>] Optionally specifies the number of seconds the PowerHub switch waits to receive a reply packet from the specified node. The default is **15** seconds. **[-size** <*pktsize*>**]** If you use the <*timeout*> argument, optionally specifies the size of the echo packet you want to send to the node. The default is **64** bytes.

Table B.4 - Bridge Subsystem Commands

Command and Description
config [show] [<*argument-list*> **| all]** Specifies the configuration parameters you want to display. The default is **all**.
bt [show] [<*seglist*> **| all] [**<*ethaddr*>**] [-t | [[-h] [-m]]]** Specifies the segment(s) for which you want to display bridge table entries. The default is **all**.
set aging [<*time*>**]off** Specifies the aging time to clear learned entries in seconds, complextime (hh:mm:ss) or tiny time (microseconds or milliseconds). Default is set to 60 minutes.
set aging [<*time*>**]off** Specifies the aging time to clear learned entries in seconds, complextime (hh:mm:ss) or tiny time (microseconds or milliseconds). Default is set to 60 minutes.
enable|disable spantree Specifies whether you are enabling or disabling the Spanning-Tree algorithm. The default is **disable**.
spantree|st set bridge-priorit|bp <*priority*> Specifies the Spanning-Tree bridge priority. The default is **8000** (hex).

Table B.4 - Bridge Subsystem Commands

Command and Description
`spantree|st sset seg-priorit|sp` *<priority> <seglist>* Specifies the Spanning-Tree segment priority. The default is **`8000`** (hex).
`spantree|st sset path-cost|pc` *<path-cost> <seglist>* Specifies the cost of the path. The default is 100 for 10Mb/s Ethernet segments, and 10 for FDDI and Fast Ethernet segments.
`spantree|st set maxage` *<time>* Specifies the maximum age, in seconds. The default is **`21`** seconds
`spantree|st set hello` *<time>* Specifies the hello time, in seconds. The default is **`4`** seconds.
`spantree|st set fwddelay` *<time>* Specifies the forward delay, in seconds. The default is **`16`** seconds.
`spantree|st set high-util` *<percentage>* Specifies the upper-end value of segment utilization. This value is a percentage in the range of **`1`** to **`100`**. The default is **`70`**%.
`spantree|st set low-util` *<percentage>* Specifies the upper-end value of segment utilization. This value is a percentage in the range of **`1`** to **`100`**. The default is **`50`**%.

Table B.5 - DECnet Subsystem Commands

Command and Description
set max-node-num | mnn *<value>* This determines the number of nodes that can exist within the PowerHub switch's area. The default is 255.

Table B.6 - Host Subsystem Commands

Command and Description
set kadelay | kad *<minutes>* Specifies how many minutes the hub allows a TCP (TELNET) connection to remain idle before sending keep-alive packets. The default is **`20`** minutes.

Table B.6 - Host Subsystem Commands

Command and Description
set kainterval | kai <*seconds*> Specifies how often the hub sends keep-alive packets before ending a connection. The default is **75** seconds.

Table B.7 - IP Subsystem Commands

Command and Description
interface add <*vlanid*> <*ipaddr*> **[**<*prefixlen*> **|** <*mask*>**] [br[oadcast] 0 | 1] [met[ric]** <*metric*>**]** Allows a standard IP subnet mask to be used. If a particular network uses IP subnet addressing, then the subnet mask should be specified here using dotted-decimal notation. Otherwise, the system uses a default subnet mask equal to the "natural" subnet mask for the particular class of address. [br[oadcast] 0 | 1 Specifies the style of broadcast address on a segment-by-segment basis. The default is **br1**. **[met[ric]** <*metric*>**]** Specifies an additional cost of using the subnet interface. The default is zero.
route enable|disable <*destination*> <*gw-ipaddr*> <*metric*> <*segment*> Specifies whether you are enabling or disabling IP routing. The default is disable.
arp set|show|unset age <*time*> <time> Specifies (in minutes) a new aging interval or turns aging off. The default is 5 minutes.
ping | pi [-t <*timeout*>**] [-size** <*size*>**]** <*ipaddr*> **[-t** <*timeout*>**]** Specifies how many seconds the PowerHub switch waits for a response from the specified device. The default is **5** seconds. **[-size** <*size*>**]** Specifies the packet length. You can specify any length from **64** through **1472** bytes. The default is **64** bytes.
ipdefaultttl|ittl set <*value*> Specifies the new TTL time in hops. The default is 16 hops.

Table B.7 - IP Subsystem Commands

Command and Description
enable|disable send-icmp-redirect|sir Specifies whether you are enabling or disabling ICMP redirect messages. The default is **enable**.
enable | disable fwd-pkts-with-srcrt-option | fps Specifies whether you are enabling or disabling source-route filtering. The default is **enable**.

Table B.8 - IP Multicast Subsystem Commands

Command and Description
it | interface add *<ipaddr>* **[met[ric]***<metric>***] [thresh[old]***<thresh>***]** **[met[ric]***<metric>***]** Specifies an additional cost (measured in hops to the destination) of using the interface. The default is **1**. **[thresh[old]***<thresh>* Specifies the minimum time-to-live (TTL) value that an IP Multicast packet must have before it is forwarded over this interface. The default is **1**.
tunnel add [-s] loc[al]*<local-addr>* **rem[ote]***<remote-addr>* **[met[ric]***<mv>* **[thresh[old]***<tv>***]** **[met[ric]***<mv>* Specifies an additional cost (extra hops to the destination) of using the virtual interface with which this tunnel is associated. The default is **1**. **[thresh[old]***<tv>***]** Specifies the minimum time-to-live (TTL) value that an IP Multicast packet must have before it can be forwarded through the tunnel. The default is **1**.
enable | disable ipm Specifies whether you are enabling or disabling IP Multicast forwarding. The default is **disable**.
penable | pdisable transmit *<segment-list>* Specifies whether you are enabling or disabling IP Multicast forwarding. The default is **penable**.

Table B.8 - IP Multicast Subsystem Commands

Command and Description
enable|disable multicast-aware-bridging Specifies whether you are enabling or disabling multicast-aware-bridging. The default is **disabled**.
it|interface add <*ipaddr*> **[met[ric]**<*metric*>**] [thresh[old]**<*thresh*>**]** **[met[ric]**<*mv*> Specifies an additional cost (measured in hops to the destination) of using the interface. The default is **1**. **[thresh[old]**<*tv*>**]** Specifies the minimum time-to-live (TTL) value that an IP Multicast packet must have before it can be forwarded over this interface. The default is **1**.
enable|disable fwd-pkts-with-srcrt-option|fps Specifies whether you are enabling or disabling source-route filtering. The default is **enable**.

Table B.9 - IP/OSPF Subsystem Commands

Command and Description
asbd enable|disable Specifies whether you want to enable or disable the PowerHub switch to function as an Autonomous System Border router. The default is **disable**.
auto-vlink enable|disable Specifies whether you want to enable or disable the automatic virtual-link feature. The default is **enable**.
area add <*area-id*> [<*auth-type*>] [stub-area-cost|sac <*cost*>] <auth-type> **md5|m** Specifies that MD5 authentication is required for OSPF packets sent within this area. The PowerHub default is none (no authentication). stub-area-cost|sac <*cost*> The OSPF software configures the default route automatically.

Table B.10 - IP / RIP Subsystem Commands

Command and Description
rip-bridging | rb [enable | disable] Enables or disables the RIP bridging feature. The default is **`disable`**.

Table B.11 - IPX Subsystem Commands

Command and Description
interface | it add *<segmentlist>* *<network>* **[mtu** *<mtu>***] [met[ric]** *<metric>***]** **[encap enet | 802.3 | 802.2 | snap]** **[mtu** *<mtu>***]** Specifies the maximum transmission unit (number of octets) for packets forwarded on this segment.
enable | disable [ipx] Specifies whether you are enabling or disabling IPX forwarding. The default is disable.
set ripsap-ctrl | rsct [normal | n vlan | v] normal | n Specifies that RIP and SAP updates are generated on a per-segment basis. This is the default.
penable | pdisable type20-port-forwarding | tpfw *<seglist>* Specifies whether you are enabling or disabling type-20 packet forwarding. The default is penable (enabled).

Table B.12 - TFTP Subsystem Commands

Command and Description
get -a fore / ph / ethan.env ethan.env Specifies net-ASCII mode. Files are transferred in binary mode by default.

Table B.12 - TFTP Subsystem Commands

Command and Description
get [-h *<host>* **] [-a]** *<remote-file>* **[** *<local-file>* **| tty]** Specifies the IP address, in dotted-decimal notation, of the TFTP server. If you do not specify this argument, the default server is used. The default server is specified using the **`set server`** command.
put [-h *<host>* **] [-a]** *<localfile>* **[** *<remote-file>* **]** Specifies the IP address, in dotted-decimal notation, of the TFTP server. If you do not specify this argument, the default server is used. The default server is specified using the **`set server`** command.
put [-h *<host>* **] [-a]** *<localfile>* **[** *<remote-file>* **]** Specifies the IP address, in dotted-decimal notation, of the TFTP server. If you do not specify this argument, the default server is used. The default server is specified using the **`set server`** command.

APPENDIX C Packet Encapsulation Formats

This appendix describes the packet encapsulations used by the PowerHub and lists the encapsulation translations performed when forwarding between Ethernet and FDDI segments.

C.1 Encapsulation Descriptions

The following sections describe the Ethernet and FDDI encapsulations used by the PowerHub bridging and routing engines.

C.1.1 Ethernet Encapsulations

The PowerHub system uses the following Ethernet encapsulation types:

- Ethernet Type II
- IEEE 802.3
- Ethernet 802.3/802.2
- Ethernet 802.3/802.2/SNAP

Above each field in the encapsulation type shown below is the size of the field in octets.

C.1.1.1 Ethernet Type II (enet)

This encapsulation is the original Ethernet format. A type field always has a value greater than 0600$_{16}$.

Ethernet Type II

6	6	2	n	4
Destination Address	Source Address	Type	Data	FCS

C.1.1.2 IEEE 802.3 (802.3)

This encapsulation is a standard format for all network traffic, established by the IEEE 802.3 standards committee. Despite the existence of this standard, much Ethernet traffic still uses the Ethernet Type II format. However, Ethernet stations running software prior to NetWare version 3.1 are often set up to use 802.3 format by default..

IEEE 802.3

6	6	2	n	4
Destination Address	Source Address	Length	Data	FCS

C.1.1.3 IEEE 802.3 with LLC Header (802.2)

This encapsulation is another standard format established by the IEEE 802.3 standards committee. This format includes an additional field for "Logical Link Control (LLC)" as established by IEEE standard 802.2. Ethernet stations running NetWare software version 3.1 or later are usually set up to use 802.2 format by default.

Ethernet 802.3/802.2

6	6	2	1	1	1	n	4
Destination Address	Source Address	Length	DSAP	SSAP	Control	Data	FCS

The values for the Control fields are:

DSAP	AA
SSAP	AA
Control	03
OUI	0
Type	0x8137

C.1.1.4 IEEE 802.3 with LLC Header and SNAP

This encapsulation is yet another standard format established by the IEEE 802.3 standards committee. A special value in the DSAP/SSAP/Control fields of the LLC header (AAAA0016) indicates that the SNAP header (OUI/Type) follows. The SNAP header indicates the type of packet, and is 0000813716 for IPX packets. This format is used mainly in networks that have non-IPX 802.3 traffic (ex: TCP/IP in 802.3 format); the SNAP header allows the IPX traffic to be distinguished from the non-IPX traffic.

.

Ethernet 802.3/802.2/SNAP

6	6	2	1	1	1	3	2	n	4
Dest. Add.	Srce. Add.	Length	DSAP	SSAP	Control	OUI	Type	Data	FCS

The values for the Control fields are:

DSAP	AA
SSAP	AA
Control	03
OUI	0
Type	0x8137

C.1.2 FDDI Encapsulations

The PowerHub system uses the following FDDI encapsulation types:

- FDDI Raw Format
- FDDI 802.2
- FDDI 802.2/SNAP

Above each field in the encapsulation type shown below is the size of the field in octets.

C.1.2.1 FDDI Raw Format

FDDI Raw Format

1	6	6	n
FC	Destination Address	Source Address	Data

C.1.2.2 FDDI 802.2

FDDI 802.2

1	6	6	1	1	1	n	4
FC	Destination Address	Source Address	DSAP	SSAP	Control	Data	FCS

C.1.2.3 FDDI 802.2/SNAP

FDDI 802.2/SNAP

1	6	6	1	1	1	3	2	n	4
FC	Dest. Add.	Srce. Add.	DSAP	SSAP	Control	OUI	Type	Data	FCS

C.1.3 Encapsulation Translations

When packets are forwarded, their encapsulations are translated according to the encapsulations specified for the source and destination networks.

C.1.3.1 Bridging

Standard bridging is used to bridge AppleTalk, DECnet, and other types of packets between Ethernet and FDDI.

In Ethernet-to-Ethernet translation, no modifications are made to the packets. However, in Ethernet-to-FDDI or FDDI-to-Ethernet translation, packets must be translated.

PowerHub software also supports IPX translation bridging. Using IPX translation bridging, you can specify the encapsulation types used by each IPX interface.

The following notes apply to these tables:

- When Ethernet II (enet) packets are on an FDDI segment after bridging, their encapsulation type is 802.2/SNAP.
- When SNAP packets are on an Ethernet segment after bridging, their encapsulation type changes to Ethernet II.
- When converting from one encapsulation type to another, the `Data` field is moved without change. Other fields are added, deleted, or modified as required.

IPX bridging over FDDI: Ethernet to FDDI to Ethernet

Ethernet format before bridging...	enet	802.3	802.3/802.2	802.2/SNAP
enet	4			
802.3		4		
802.3/802.2			4	

IPX bridging over FDDI: Ethernet to FDDI

Ethernet format before bridging...	802.3	802.2	802.3/802.2	802.2/SNAP
enet				4
802.3	4			
802.3/802.2		4		

IPX bridging over FDDI: FDDI to Ethernet

FDDI format before bridging...	802.3	802.2	enet
802.3	4		
802.3/802.2		4	

C.1.3.2 Routing

IP packets are translated between Ethernet and FDDI format as required by RFC 1103, using Ethernet headers on the Ethernet side and 802.2/SNAP headers on the FDDI side. (The difference is that FDDI format contains 802.2 LLC and SNAP headers before the IP header, where the SNAP header contains a type value corresponding to the type field in the Ethernet transmission.)

AppleTalk packets retain their 802.3 headers, even on the Ethernet side, in order to preserve the SNAP header used by the AppleTalk protocol.

C.1.3.3 Maximum Transmission Unit (MTU) Discovery

When FDDI packets are forwarded to Ethernet segments or Ethernet packets are forwarded to FDDI segments, the PowerHub software changes the packets into an appropriate frame type before forwarding them.

FDDI IP packets larger than 1518 bytes are fragmented, as specified by RFC 791, whether they are bridged or routed.

For routed traffic, maximum transmission unit (MTU) discovery is implemented in accordance with RFC 1191. When you configure an IP interface (using the **`ip add-interface`** command), the PowerHub software automatically sets the MTU value for IP interfaces based on the medium type:

- For interfaces on FDDI segments, the MTU is set to 4050.
- For interfaces on Ethernet segments, the MTU is set to 1500.
- If the interface spans multiple segments, and those segments include both Ethernet and FDDI, the MTU value is set to 1500.

APPENDIX D Netboot Options

This appendix describes the netbooting process in detail and describes how you can share a common boot definition file among multiple PowerHubs. For additional netbooting options information, see the *PowerHub Hardware Reference Manual*.

The PowerHub implementation of netbooting uses the Boot Protocol (BOOTP) and Trivial File Transfer Protocol (TFTP). PowerHub netbooting is designed to be fully compliant with RFCs 951, 1048, and 1350. The PowerHub can netboot over any type of Ethernet or Fast Ethernet segment, and ATM, but not over FDDI.

After configuring the PowerHub for netbooting, the netboot process can begin by booting (or rebooting) the system, using any of the following methods:

- Press the reset switch (labeled RST), located on the front of the Packet Engine.
- Issue the **reboot** command.
- Issue the **boot** (**b**) command at the <PROM-7pe2> prompt.
- Turn the power supply off, then back on.

D.1 Choosing a Netbooting Method

The boot process differs depending upon whether the client PowerHub and server are on the same subnet or different subnets. Accordingly, the netbooting implementation depends upon the network configuration.

Point-to-point netbooting can be used if the client PowerHub and the BOOTP/TFTP server are on the same subnet. The subnet can be a single segment or multiple segments connected by bridges. Point-to-point netbooting is the simplest to implement. It is recommend to use point-to-point netbooting when the client PowerHub and the BOOTP/TFTP server are all on the same subnet.

If the client PowerHub and server are on different subnets, do one of the following:

- Implement a boot helper service on the client PowerHub subnet.The PowerHub provides a service called *IP Helper* that can forward UDP packets (including BOOTP packets) between a netboot client and a remote server.

- Manually insert information (such as the client and server IP addresses and the IP address of the gateway) into the NVRAM. NVRAM contains a battery backup and retains its data across power cycles. The contents of the NVRAM are not lost, even if the system is powered down.

The boot parameters configured in NVRAM override the corresponding parameters returned by the BOOTP server. If the PowerHub is to bypass the BOOTP process, configure all the applicable boot values in NVRAM.

IPX Router Indicates whether main memory has been allocated for the IPX subsystem.

IPX Forwarding Indicates whether IPX forwarding is enabled or disabled. The default setting is disabled.

IPX Type20 Packet Forwarding Indicates that the switch is configured to forward type-20 IPX packets. The default setting is enabled.

D.2 The Boot Process

The netbooting process takes place in the following phases:

BOOTP BOOTP packets are exchanged. (The BOOTP phase is bypassed if applicable boot parameters are configured in NVRAM.)

BOOTDEF Boot definition file is received via TFTP from server and parsed. The boot definition file specifies the configuration file and system software to be used.

IMAGE Image files (system software) are received via TFTP from server and executed.

CONFIG Configuration file is received via TFTP from server and executed.

RUN-TIME Normal run-time operation begins.

The last four phases are identical for each netbooting implementation. However, the first phase (BOOTP) differs according to the implementation. For reference, the tables in the following sections summarize the netbooting process used by each method of netbooting. It is not necessary to know the netbooting phases in detail to implement netbooting, but these tables can help to troubleshoot the netbooting implementation.

D.2.1 Point-to-Point

The following table summarizes the netbooting process used when the PowerHub and BOOTP server are on the same subnet.

Table D.1 - Point-to-Point netbooting

Phase	Process
BOOTP	• BOOTP broadcast packet sent out each Ethernet segment. The BOOTP packet contains the MAC address, but no other address information. • Server receives broadcast packet and sends BOOTP reply packet (provided the MAC address in the bootptab file, or equivalent, on the BOOTP server). Reply packet contains server's IP address, IP address, IP subnet mask, and name of boot definition file. • BOOTP response received and information stored from server in memory. During the boot process, each Ethernet segment is configured as an IP interface by default. This segment configuration has no relation to the configuration of the segments during run-time operation.
BOOTDEF	• TFTP used to transfer boot definition file from server. • Boot definition file parsed. • While parsing boot definition file, names of image files and configuration file obtains and stored in memory.
IMAGE	• TFTP used to transfer image files from server and load them into memory and executes them.
CONFIG	• Name of the configuration file retrieved from memory. • Interface (segment) received from BOOTP reply and interface configured for TFTP exchanges. • TFTP used to transfer configuration file from server and saves file in memory. • Interface de-configured. • Configuration file executed.
RUN-TIME	• Normal bridging and routing according to settings in configuration file.

D.2.2 Cross Gateway (Boot Helper Service Used)

The following table summarizes the netbooting process used when the PowerHub and BOOTP server are on separate subnets, and a boot helper service (such as IP Helper) is implemented. The intervening gateway that connects the segments can be another PowerHub or any other device that implements a boot helper service.

Table D.2 - Helper-Assisted Netbooting

Phase	Process
BOOTP	• PowerHub sends a BOOTP broadcast packet out each Ethernet segment. The BOOTP packet contains MAC address, but no other address information. • BOOTP request is received by intervening gateway on a segment previously configured with an IP Helper address. • IP Helper facility in intervening gateway forwards BOOTP packet to server. • Server receives BOOTP request forwarded by intervening gateway and sends response packet to gateway. Response packet contains name of boot definition file, server's IP address, PowerHub IP address, PowerHub IP subnet mask, and intervening gateway's IP address. • Gateway forwards response packet to client switch. • Client switch receives BOOTP response and stores information from server in memory. During the boot process, each Ethernet segment is configured as an IP interface by default. This segment configuration has no relation to the configuration of the segments during run-time operation.
BOOTDEF	• Identical to point-to-point process.
IMAGE	• Identical to point-to-point process.
CONFIG	• Identical to point-to-point process.
RUN-TIME	• Identical to point-to-point process.

D.2.3 Cross-Gateway (No Boot Helper Service Used)

The following table summarizes the netbooting process used for cross-gateway netbooting when the gateway does not have a boot helper service. If you prefer, you can implement this method even if the intervening gateway does contain a boot helper service.

Table D.3 - Cross-Gateway Netbooting — No Boot Helper Service

Phase	Process
BOOTP	• PowerHub uses boot parameters in NVRAM as substitute for BOOTP parameters. The following parameters can be specified in NVRAM: PowerHubs IP address and subnet mask, gateway's IP address, server's IP address, name of the PowerHub boot definition file. The boot definition file contains the file names and pathnames of the software image files and configuration file. Unless all BOOTP parameters were supplied from NVRAM, PowerHub sends a BOOTP broadcast packet out each Ethernet segment configured as IP interface. Parameters not configured in NVRAM are sought in the response from the BOOTP server. • BOOTP request is received by intervening gateway. If boot parameters in NVRAM include information needed by gateway to forward the BOOTP packet, the packet is forwarded to server. This information includes the PowerHub IP address and subnet mask and the server's IP address. • Server receives BOOTP request forwarded by intervening gateway and sends response packet, through the gateway, to the PowerHub. • PowerHub receives BOOTP response and stores information from server, including name of boot definition file, in boot PROM. During the boot process, each Ethernet segment is configured as an IP interface by default. This segment configuration has no relation to the configuration of the segments during run-time operation.
BOOTDEF	• Identical to point-to-point process.
IMAGE	• Identical to point-to-point process.
CONFIG	• Identical to point-to-point process.
RUN-TIME	• Identical to point-to-point process.

D.3 Configuration Options

This section describes the configuration requirements for the BOOTP server, TFTP file server, and PowerHub for point-to-point netbooting. Implement this type of netbooting if the PowerHub, BOOTP server, and TFTP server are all attached to the same subnet. The subnet can be a single segment or multiple segments connected by bridges.

D.3.1 TFTP Server

Regardless of the netbooting method that is chosen, perform the following configuration tasks for the TFTP server (even if the BOOTP server and TFTP server are the same device):

- Install the system software image files.
- Edit and install the boot definition file(s). A separate boot definition file can be installed for each PowerHub or boot definition macros can be used to share a single boot definition file among multiple PowerHubs.
- For each PowerHub system, install its configuration file.

These files can be installed in the TFTP home directory or set up subdirectories. If subdirectories are set up, make sure the pathnames are specified in the boot definition file.

D.3.2 BOOTP Server

Configure the same host device as both a BOOTP server and a TFTP server, or configure separate BOOTP and TFTP servers.

NOTE

Although BOOTP and TFTP services can be provided by different hosts, using the same host results in faster booting because the PowerHub does not need to search across its interfaces multiple times for a server. In fact, some BOOTP servers do not support the file service from another host, so in such cases a choice is not available.

Unless all the required values into are configured in NVRAM, configure the BOOTP server to provide the following information to the PowerHub (even if the BOOTP server and TFTP server are the same device):

- Client switch's IP address.
- Client switch's subnet mask.
- Gateway's IP address (if the client switch and server are on different subnets).
- TFTP server's IP address.
- Name of the boot definition file (often called bootdef) used to boot the PowerHub. Install this file on the TFTP server, but specify the name on the BOOTP server in NVRAM. Note that the boot definition file is neither the image file (7pe) nor a configuration file (such as cfg).

The procedures for configuring the BOOTP server depend upon the BOOTP software being used. In some BOOTP software, a single database file contains the information items listed above for each client switch that uses the server. In some implementations, this file is called the bootptab file. See the BOOTP software documentation for information.

D.3.3 Intervening Gateway

If a gateway separates the PowerHub from the server, do one of the following:

- If the gateway has a boot helper service, such as IP Helper, configure the helper service to help BOOTP packets sent from the PowerHubh to reach the BOOTP server. If the gateway is another PowerHub, use the **`ip add-helper`** command.
- If the gateway does not have a boot helper service, configure the following values in NVRAM:
- Client switch's IP address.
- Client switch's subnet mask.
- Gateway's IP address (if on different subnets).
- TFTP server's IP address.
- Name of the boot definition file.

D.3.4 Client PowerHub

To configure the PowerHub for netbooting:

- Specify the boot order in NVRAM. Do this regardless of the type of netbooting implemented.
- If needed, configure boot parameters in NVRAM. See the previous section.

A *boot definition file* contains instructions for loading the system software files and the configuration file used by a PowerHub when it boots. This section describes boot definition files and boot definition macros, then tells how to edit and copy boot definition files and configuration files onto the TFTP server.

The TFTP server must contain at least one boot definition file. Edit and install a separate boot definition file for each PowerHub, or share a single boot definition file among multiple PowerHubs. The installed PowerHub software contains a boot definition file called bootdef. This bootdef file supports booting from the Flash Memory Module or Compact Flash Card. It can be copied and modified for netbooting.

Here is an example of the bootdef file that is shipped with the PowerHub 7000[1]:

```
%vstart 1
7pe2             m
%vend 1
```

To prepare a bootdef file for netbooting, copy the file shipped with the PowerHub onto the TFTP server, then modify the file as follows:

- Add or modify a line to load the configuration file. (If the configuration file has been saved before copying the bootdef file, the bootdef file already contains a line for loading the configuration file. This line needs to be modified.)
- Add the pathname and file name for the software image on the TFTP server.

Here is an example of a bootdef file that is modified for netbooting:

```
%vstart 1
fore/ph/configs/0000EF014A00.cfg     c
fore/ph/images/7-2.6.3.0/7pe         m
%vend 1
```

In this example, the PowerHub’s MAC-layer hardware address is used as the configuration file name. The pathnames for the configuration file and the software image file are included with the file names. Whether a pathname is specified depends upon how the TFTP server is configured. When editing the bootdef file, make sure the pathnames are entered that are meaningful to the TFTP server.

[1] Some boot definition files might contain the lines %vstart 0 and %vend 0. These lines are used for booting from a floppy diskette (PowerHub 7000 only) and do not work for netbooting. Make sure the boot definition files used for netbooting use %vstart 1 and %vend 1.

D.3.5 Using the Same Boot Definition File with Multiple Switches

If only one PowerHub needs to be configured for netbooting, using the PowerHub's MAC-layer hardware address to name the configuration file is a simple way to name the file. However, if more than one is being configured for network booting, do one of the following:

- Create a unique boot definition file for each PowerHub. If this method is chosen, use the **nvram set netbdfile** command to set the boot definition file in each PowerHub's NVRAM. Otherwise, each PowerHub attempts to use the default boot definition file name (bootdef).
- Use a single boot definition file, but use boot definition macros in place of the configuration file name. A *boot definition macro* is a 2-character sequence consisting of a '$' followed by a single letter. The macros are expanded by the PowerHub Packet Engine boot PROM when it reads the boot definition file. Table B–4 lists the boot definition macros.

Table D.4 - Boot definition macros

Phase	Process
$E	ASCII representation of the MAC-layer hardware address; for example, "0000EF014A00". $E always expands to 12 characters.*
$e	ASCII representation of the three least significant octets of the MAC-layer hardware address. $e always expands to 6 characters.*
$D	Directory part of the path name of the boot definition file.
$B	Base name of the boot definition file (the directory part of the path name and anything following the rightmost dot of the file name are removed).
$$	Expands to a single '$' character. Use this if using the '$' character with a boot definition macro.
***$E and $e expand hex digits A-F in uppercase**	

Here is an example of a boot definition file that uses boot definition macros.

```
%vstart 1
$D/$E.cfg                               c
fore/ph/images/7-2.6.3.0/7pe            m
%vend 1
```

In this example, the $D macro expands into the pathname of the boot definition file. The $E macro expands into the MAC-layer hardware address. When that Packet Engine parses the macros in the boot definition file, it expands $D into fore/ph/configs/ and $E into 0000EF014A00.

D.3.6 Sharing Methods

If a common boot definition file is shared among multiple PowerHubs, decide on one of the following sharing methods:

MAC-address Each configuration file is named according to the following convention: <*MAC-addr*>**.**cfg, where <*MAC-addr*> is the MAC-layer hardware address of the PowerHub.

Link On boot servers that support symbolic links to files, give meaningful names to configuration files according to the following pattern: <*name*>**.**cfg, where <*name*> is an arbitrary name assigned to the PowerHub.

These methods are very similar. They differ only in that any name can be used for the configuration files if the link method is used. However, the MAC-layer hardware addresses must be used in the configuration file names if the MAC-address method is used.

NOTE To use the link method, the TFTP server must support symbolic links. To determine if the server supports symbolic links, see the server's documentation.

The following sections contain examples of each sharing method.

D.3.6.1 MAC-Address Method

Here is an example of a TFTP server directory and file structure used to implement the MAC-address sharing method.

```
fore

    ph
        images
            7-2.6.3.0
                7pe

            7-2.6.3.1
                7pe

            7-2.6.3.2
                7pe

        configs
            bootdef-7-2.6.3.0
            bootdef-7-2.6.3.1
            bootdef-7-2.6.3.2

            014A00.cfg
            015AD0.cfg
            015AE0.cfg
            015AF0.cfg
            016AD0.cfg
```

This example shows TFTP subdirectories, but you can just as easily install the files into the TFTP home directory. As shown in this example, the configs subdirectory contains a single boot definition file for each system software version, but a separate configuration file for each PowerHub that uses this TFTP server. Each configuration file is named after the last six hexadecimal digits of a PowerHub's MAC-layer hardware address.

Recall that the boot definition file contains the name of the configuration and system software image files to be loaded. The BOOTP server tells the PowerHub which boot definition file to use. The boot definition macros are expanded to form the unique name of the configuration file for that PowerHub.

To name configuration files according to the MAC-address sharing method, use the following procedure for each PowerHub that uses the server.

For each configuration file, copy the file onto the server:

<*MAC-addr*>.cfg

<MAC-addr>.cfg Is the MAC-layer hardware address (in hex double-digit format) of the client switch. You can specify the full hardware address (12 hex digits) or, for systems that do not support long file names, the last three octets (six hex digits) of the hardware address.

NOTE Hex digits A-F **must** be in uppercase.

When this procedure is complete, the TFTP server should contain a separate configuration file for each PowerHub that uses the server to netboot. Here is an example of a TFTP server directory and file structure used to implement the link sharing method.

```
fore

    ph
        images
            7-2.6.3.0
                7pe

            7-2.6.3.1
                7pe

            7-2.6.3.2
                7pe

        configs
            bootdef-7-2.6.3.0
            bootdef-7-2.6.3.1
            bootdef-7-2.6.3.2

            ph-1.bd
            ph-2.bd
            ph-3.bd

            ph-1.cfg
            ph-2.cfg
            ph-3.cfg
```

This example shows TFTP subdirectories, but you can just as easily install the files into the TFTP home directory. As shown in this example, the configs subdirectory contains a single boot definition file for each version of runtime software, and a configuration file for each client PowerHub that uses this TFTP server. It also contains a link to each configuration file. Each link has the same base name as the corresponding configuration file, but has the extension.bd. This base name is associated with a particular client switch by a bootptab file or other file containing IP information and other information needed by your BOOTP software. See the documentation for your file server to determine how to associate the link names with the configuration files.

To name your configuration files according to the link sharing method, use the following procedure for each client switch that will use the server.

For each configuration file, copy the file into the `/fore/ph/configs` directory as:

```
<name>.bd
```

<name>.bd A meaningful name that rpresents the file. Any file name that is legal on your server can be used.

When you complete this procedure, your TFTP server should contain a separate configuration file for each client switch that will use the server.

APPENDIX E Well-Known Ports

This appendix lists the well-known names provided in RFC 1340 that the PowerHub system supports. When configuring a PowerHub IP or TCP filter, you can supply either the port number or the well-known name to specify the destination port of packets that you want to either block or accept. You supply the port number or well-known name in the <*dstseg*> field of templates for any TCP and IP filters you create. The <*dstseg*> field is used with the following TCP and IP filter commands:

- tcp tcp-filter add
- ip ip-fil-acs-ctrl add

When an IP packet comes in on an Ethernet segment, the Ethernet header is stripped away. The packet then relies on the IP header to begin routing it through your LAN to its eventual destination. In the IP header, the protocol type field denotes the kind of packet that follows, such as ARP, TCP, or UDP.

If the protocol type field indicates a TCP or UDP packet, then that packet is travelling from a source port to a destination port; a 16-bit number represents each port. Many of these ports are considered “well-known” ports because they appear in an official, published table (RFC 1340) that relates the names of commonly-used protocols with the TCP or UDP ports they typically use.

Appendix E.1 lists alphabetically the well-known names that the PowerHub system recognizes, and provides the port number associated with each well-known name. Enter the “well-known” port name or number exactly as shown in the table.

Table E.1 - Well Known Names and Ports

Well-known Name	Port Number	Well-known Name	Port Number
`at-echo`	204	`at-nbp`	202
`at-rtmp`	201	`at-zis`	206
`auth`	113	`bgp`	179
`biff`	512	`bootpc`	68
`bootps`	67	`chargen`	19
`courier`	530	`csnet-ns`	105

Table E.1 - Well Known Names and Ports

Well-known Name	Port Number	Well-known Name	Port Number
daytime	13	discard	9
dls	197	domain	53
echo	7	exec	512
finger	79	ftp	21
ftp-data	20	hostname	101
hostnames	101	ingreslock	1524
ipcserver	600	ipx	213
iso-tp0	146	isop-tsap	102
kerberos	88	klongin	543
kshell	544	link	87
login	513	lpd	515
monitor	561	nameserver	42
netbios-dgm	138	netbios-ns	137
netbios-ssn	139	netwews	532
netstat	15	news	144
NeWs	144	new-rwho	550
nicname	43	nntp	119
npp	92	ntp	123
pcserver	600	pop-2	109
pop3	110	printer	515
print-srv	170	rip	520
rlogin	513	rmonitor	560
route	520	rtelnet	107
rwho	513	shell	514
smtp	25	snmp	161
snmptrap	162	sunrpc	111
supdup	95	syslog	514

Table E.1 - Well Known Names and Ports

Well-known Name	Port Number	Well-known Name	Port Number
systat	11	tacnews	98
talk	517	tcpmux	1
telnet	23	tftp	69
time	37	timed	525
uucp	540	who	513
whois	43	x400	103
x400-snd	104	xdmcp	177
supdup	95	syslog	514

Index